

TDX-PITE 型

微机系统与接口

实验讲义

(2024 版)

吉林大学计算机科学与技术学院

2024 年 8 月

前　　言

《微机系统与接口实验》是一门实践性和综合性很强的实验课，是对《数字逻辑电路》、《计算机组成原理》、《微机技术》等前期硬件课程学习成果的检验和综合运用。

1、教学目的

- (1) 加深对微机基本逻辑部件的工作原理和基本应用方法的理解；
- (2) 加深对微机各功能部件工作过程的理解，如 RAM 的读写过程，中断的请求、允许和响应过程，DMA 的工作过程等；
- (3) 加深对微机接口的理解并掌握微机管理外部设备的基本原理和方法；
- (4) 掌握 82 系列芯片，以及一些常用部件（如小键盘、晶体灯、LED 数码管等）的基本应用方法；
- (5) 掌握小型微机应用系统的设计和开发方法。

总之，通过本课程的学习与实践，可以加深学生对微机系统各功能部件工作过程的理解，掌握 CPU 及其外围接口芯片的性能和使用方法，加强学生硬件基本功的训练，培养学生的计算机应用与开发能力。

2、实验要求

- (1) 课前应认真阅读实验讲义，熟悉实验设备的使用方法及注意事项；
- (2) 实验前做好预习准备工作，理解实验原理，编写实验程序；
- (3) 实验过程中按要求连接实验板上的电路，调试程序，认真观察、分析实验结果；
- (4) 实验完成后需经指导教师检查验收；
- (5) 实验结束后按要求及时书写实验报告并按时上交。实验报告内容包括：实验名称、实验目的、实验内容与方法、程序框图、程序代码、实验结果分析与体会。实验报告字迹要工整、清晰。

注意事项：

- 要严格按操作规程操作，严禁带电插拔连线；
- 开机前一定要检查连线，确保正确无误方可通电；
- 因违反操作规程而损坏实验设备者，按学校有关规定赔偿。

《微机系统与接口实验》评分标准

1、本实验课成绩分为：优秀、良好、中等、及格和不及格。

2、评分依据：

$$\text{实验课成绩} = \text{平时成绩} + \text{上机考试成绩} - \text{纪律扣分}$$

(1) 平时成绩：40 分，具体见下表。

| 实验名称 | 分数 | 实验报告分数 |
|---------------|----|--------|
| 8255 并行接口实验 | 2 | 0 |
| 存储器扩展实验 | 3 | 0 |
| 8259 中断优先级实验 | 3 | 2 |
| A/D 转换实验 | 3 | 0 |
| D/A 转换实验 | 3 | 0 |
| 8254 定时/计数器实验 | 3 | 2 |
| 8251 串行接口实验 | 3 | 0 |
| 数码管显示实验 | 3 | 0 |
| 键盘扫描及数码管显示实验 | 3 | 2 |
| 微机接口综合实验 | 5 | 0 |
| 合计 | 34 | 6 |

(2) 上机考试成绩：60 分。考试成绩按整个考场完成同一考题内容的时间先后顺序评定。

(3) 纪律扣分

- 旷课一次，扣 10 分；累计旷课 3 次以上，本课成绩为不及格。迟到一次，扣 2 分；迟到 15 分钟以上，视为旷课。
- 违反操作规程损坏实验设备者，除按学校有关规定赔偿外，扣 10 分。
- 应独立完成实验。如有抄袭，则抄袭一个实验扣 5 分。
- 保持实验室清洁、肃静，严禁在实验室吸烟、大声喧哗、乱串座位、乱扔纸屑、接听手机。屡犯不改者扣 5 分。

目 录

| | |
|-----------------------------------|-----------|
| 第一章 TDX-PITE 实验系统硬件概述..... | 1 |
| 1.1 系统结构..... | 1 |
| 1.2 系统硬件..... | 1 |
| 1.3 80X86 微机系统单元 | 2 |
| 1.3.1 系统总线..... | 2 |
| 1.3.2 系统中的 8259 单元..... | 2 |
| 1.4 接口实验单元..... | 2 |
| 1.4.1 SRAM 实验单元..... | 2 |
| 1.4.2 8237DMA 实验单元..... | 3 |
| 1.4.3 8259 实验单元..... | 4 |
| 1.4.4 8254 定时/计数器单元..... | 4 |
| 1.4.5 8255 并行接口单元..... | 5 |
| 1.4.6 8251 串行通信单元 | 5 |
| 1.4.7 模/数转换单元..... | 6 |
| 1.4.8 数/模转换单元..... | 7 |
| 1.4.9 键盘扫描及数码管显示单元..... | 7 |
| 1.4.10 点阵 LED 显示单元..... | 8 |
| 1.4.11 驱动电路单元..... | 8 |
| 1.4.12 步进电机与直流电机单元 | 9 |
| 1.4.13 电子发声单元与温度控制单元..... | 9 |
| 1.4.14 单次脉冲单元..... | 9 |
| 1.4.15 逻辑开关及 LED 显示单元..... | 10 |
| 1.4.16 地址译码单元 | 10 |
| 1.4.17 16 位 I/O 接口单元..... | 11 |
| 1.4.18 转换单元..... | 11 |
| 1.4.19 扩展单元..... | 11 |
| 1.4.20 系统总线单元 | 12 |
| 1.5 系统电源..... | 12 |
| 第二章 TDX-PITE 系统软件概述..... | 13 |
| 2.1 TDX-PTE 联机软件使用说明 | 13 |
| 2.1.1 主窗口界面..... | 13 |
| 2.1.2 菜单功能 | 15 |
| 2.1.3 工具栏功能介绍..... | 22 |
| 2.1.4 专用图形显示 | 23 |
| 2.1.5 示波器..... | 25 |
| 2.1.6 时序观测 | 26 |
| 2.1.7 自定义功能 | 27 |
| 2.1.8 右键菜单功能 | 31 |
| 2.1.9 Debug 调试命令..... | 31 |
| 2.2 系统编程信息..... | 34 |
| 2.2.1 地址分配情况 | 34 |

| | |
|---|----|
| 2.2.2 常用 BIOS 及 DOS 功能调用说明..... | 35 |
| 附录 1 INTEL 386 TM EX 嵌入式处理器框图..... | 37 |
| 附录 2 FPGA 板介绍..... | 38 |
| 第三章 实验内容..... | 41 |
| 实验一 8255 并行接口实验 | 41 |
| 实验二 存储器扩展实验 | 44 |
| 实验三 8259 中断优先级实验 | 48 |
| 实验四 A/D 转换实验 | 54 |
| 实验五 D/A 转换实验 | 57 |
| 实验六 8254 定时/计数器实验 | 59 |
| 实验七 8251 串行接口实验 | 63 |
| 实验八 数码管显示实验 | 68 |
| 实验九 键盘扫描及数码管显示实验 | 71 |
| 实验十 微机接口综合实验 | 72 |

第一章 TDX-PITE 实验系统硬件概述

1.1 系统结构

TDX-PITE 实验系统自带 Intel 386EX 微处理器，使用 RS-232 串口与微机相连，通过串口通讯完成微机接口硬件实验。TDX-PITE 实验系统结构如表 1-1 所示。

表 1-1 TDX-PITE 实验系统结构

| | |
|------------|---|
| I386EX 单板机 | Intel i386EX、Flash ROM（128KB）、SRAM（128KB） |
| 信号源 | 单次脉冲：消抖动脉冲2组 |
| 逻辑电平开关与显示 | 16组电平开关，16组电平显示LED灯（正逻辑） |
| 接口实验单元 | 8259、8237、8254、8255、8251、TLC7528、ADC0809、SRAM、键盘扫描及数码管显示、扬声器、点阵LED、直流电机、步进电机、温度控制 |
| 实验扩展单元 | 2组40线通用集成电路扩展单元、扩展模块总线单元、FPGA模块 |
| 系统电源 | +5V |

1.2 系统硬件

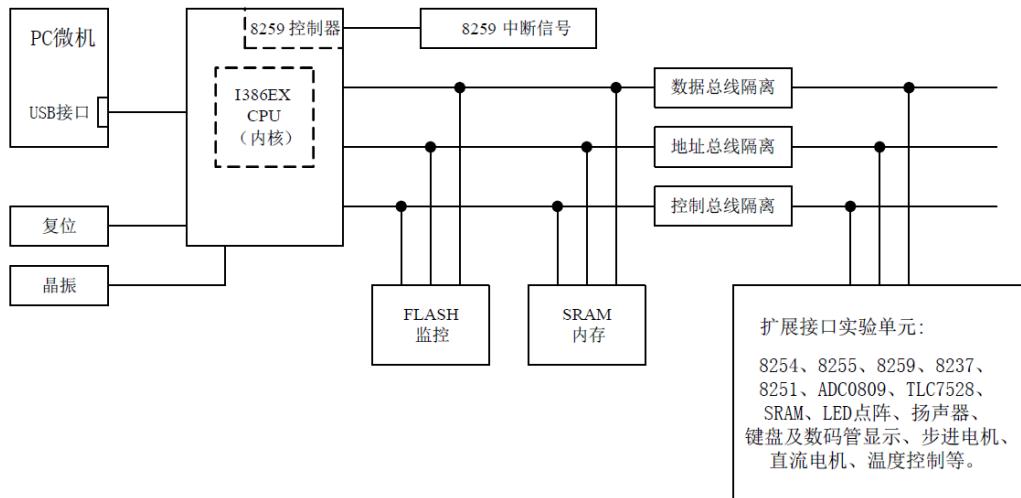


图 1-1 TDX-PITE 系统硬件结构图

1.3 80X86 微机系统单元

1.3.1 系统总线

TDX-PITE 采用组合式结构，即 I386EX 系统板加实验接口平台的形式。将 I386EX 系统板扣在实验接口平台上便构成 80X86 微机原理及接口技术教学实验系统，系统总线以排针和锥孔两种形式引出，实验时，与实验单元相连可完成相应的实验。系统引出信号线说明见表 1-2。

表 1-2 80X86 微机系统信号线

| 信号线 | 说 明 | 信号线 | 说 明 |
|-------------|-------------------|-------------|-------------------|
| XD15 ~ XD0 | 系统数据线（输入/输出） | MIR6、MIR7 | 主 8259 中断请求信号（输入） |
| XA19 ~ XA1 | 系统地址线（输出） | MWR #、MRD # | 存储器读、写信号（输出） |
| BHE #、BLE # | 字节使能信号（输出） | IOW #、IOR # | I/O 读、写信号（输出） |
| ADS # | 地址状态信号（输出） | RST # | 复位信号（输出） |
| MY0、MY1 | 存储器扩展用片选信号（输出） | CLK | 1MHz 时钟输出 |
| IOY0 ~ IOY3 | I/O 接口扩展用片选信号（输出） | 1.8432MHz | 1.8432MHz 时钟输出 |
| HOLD | 总线请求（输入） | 184.32KHz | 184.32KHz 时钟输出 |
| HLDA | 总线允许应答（输出） | 18.432KHz | 18.432KHz 时钟输出 |
| INTA# | 中断响应信号（输出） | | |

注：# 表示该信号低电平有效

1.3.2 系统中的 8259 单元

I386EX 单板机中提供有主片 8259 中断控制器，该主片的中断请求信号 IR6 和 IR7 对实验平台开放，供实验使用。实验平台上的 8259 单元可以用于级联方式实验，关于这部分的内容详见 8259 实验部分。

1.4 接口实验单元

每个单元的电源与地均已连接好，“圆圈”表示该信号通过排针或锥孔引出，实验中需要通过排线或弹簧线进行必要的连线来完成实验。

1.4.1 SRAM 实验单元

SRAM 实验单元由两片 6264 组成 $8K \times 16$ 的存储器访问单元，数据宽度为 16 位，低字节与高字节的选择由 BLE、BHE 决定。如果只需要使用一片 $8K \times 8$ 的存储器时，可以将 BLE 信号直接与 GND 相接。电路图如图 1-2 所示。

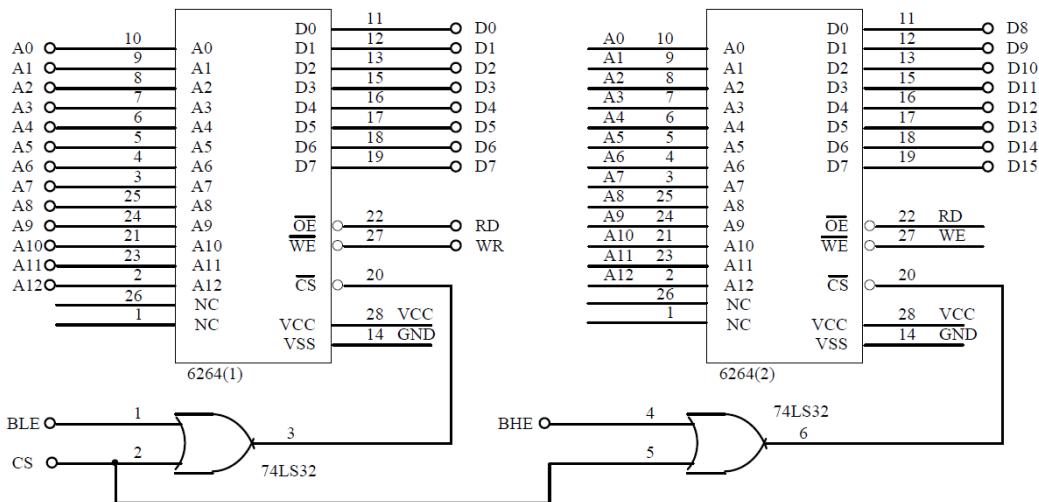


图 1-2 SRAM 实验单元电路

1.4.2 8237DMA 实验单元

DMA 实验单元主要由 8237 和 74LS573 组成，如图 1-3 所示。

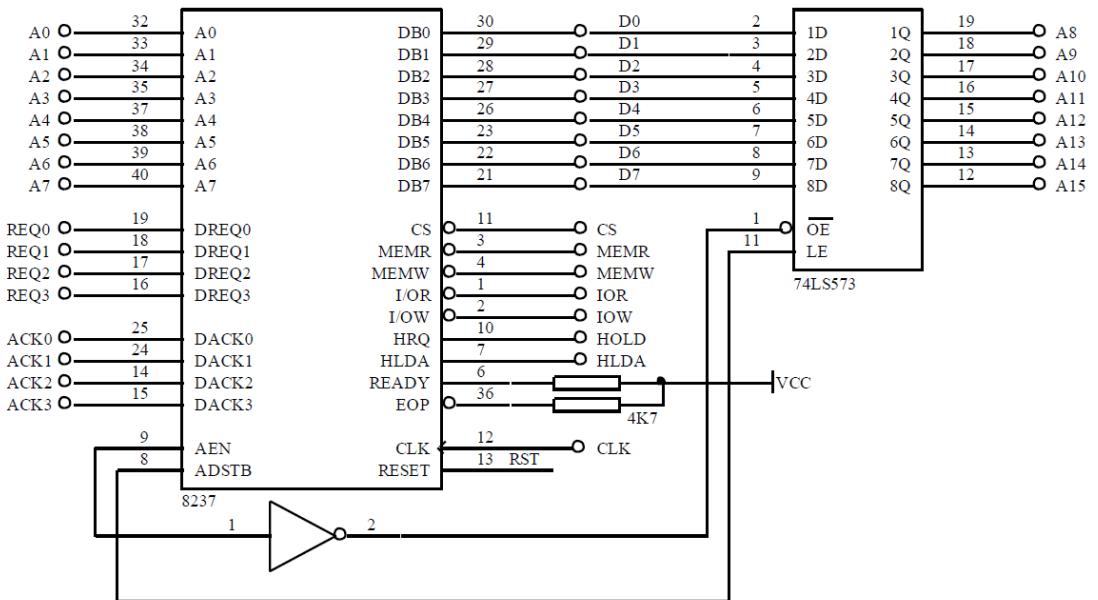


图 1-3 8237 DMA 实验单元电路

1.4.3 8259 实验单元

8259 实验单元由 8259 组成，如图 1-4 所示。

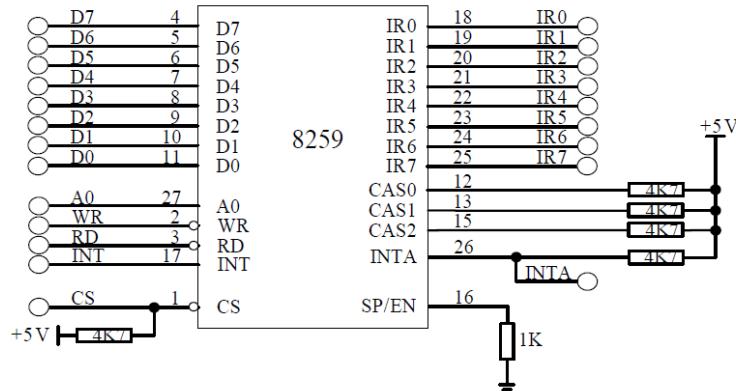


图 1-4 8259 实验单元电路

1.4.4 8254 定时/计数器单元

8254 共有三个独立的定时/计数器，其中，0 号和 1 号定时/计数器开放出来可任意使用，2 号定时/计数器用于为 8251 串行通信单元提供收发时钟。2 号定时/计数器的输入为 1.8432MHz 时钟信号，输出连接到 8251 的 TxCLK 和 RxCLK 引脚上。定时/计数器 0 的 GATE 信号连接好了上拉电阻，若不对 GATE 信号进行控制，可以在实验中不连接此信号。具体实验电路图如图 1-5 所示。

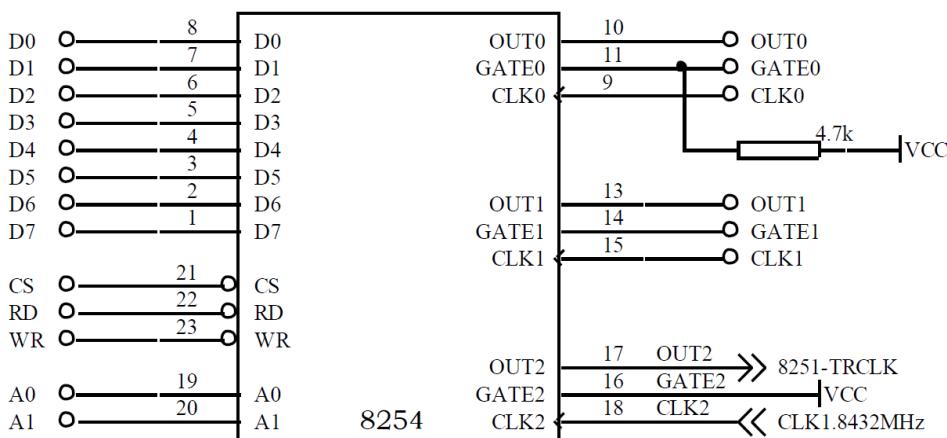


图 1-5 8254 定时/计数器单元电路

1.4.5 8255 并行接口单元

并行接口单元由 8255 组成，其复位信号已连接到系统复位上，如图 1-6 所示。

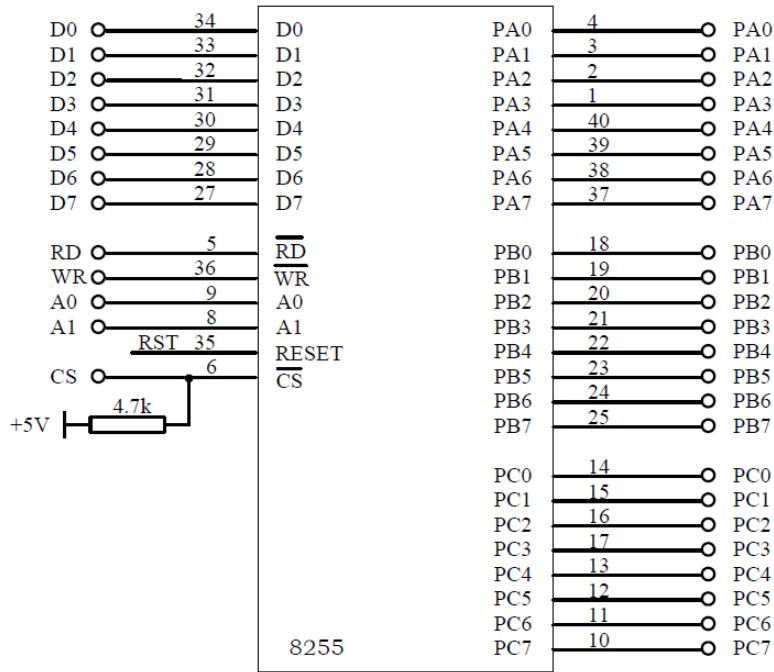


图 1-6 8255 并行接口实验单元电路

1.4.6 8251 串行通信单元

串行通信控制器选用 8251，收发时钟来自于 8254 单元的定时/计数器 2 的输出，控制器的复位信号已与系统连接好，如图 1-7 所示。

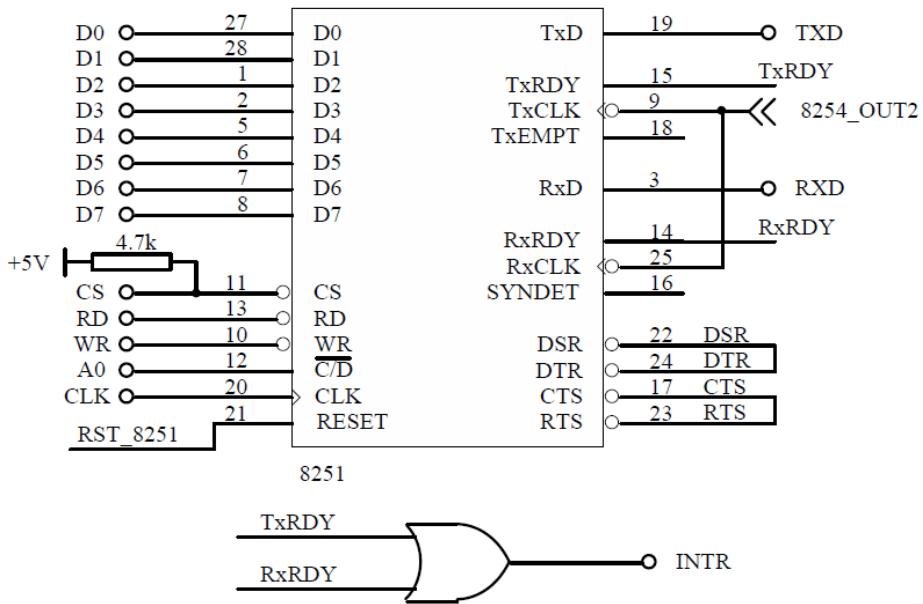


图 1-7 8251 串行通信实验单元电路

1.4.7 模/数转换单元

模/数转换实验单元由ADC0809 芯片及电位器电路组成。ADC0809 的 IN7 通道用于温度控制实验，如图 1-8 所示。

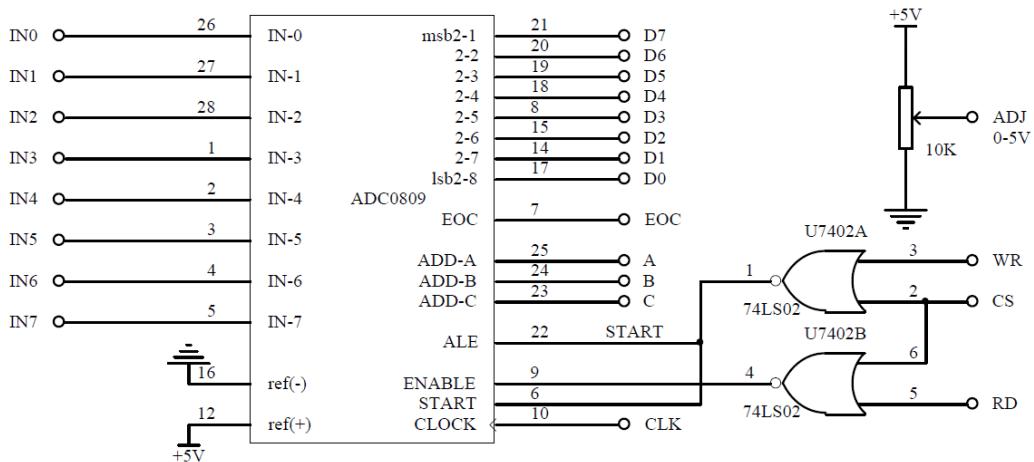


图 1-8 模/数转换实验单元电路

1.4.8 数/模转换单元

D/A 转换实验单元由 TLC7528 与 LM324 构成，电路图如图 1-9 所示。

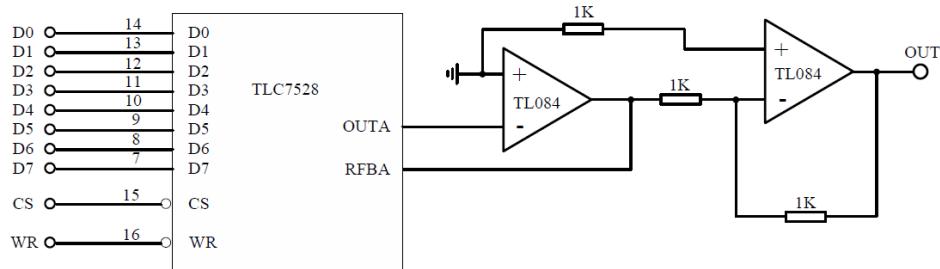


图 1-9 数/模转换实验单元电路

1.4.9 键盘扫描及数码管显示单元

键盘扫描与数码管显示单元由 6 个共阴极数码管、 4×4 键盘扫描阵列及显示驱动电路组成，如图 1-10 所示。

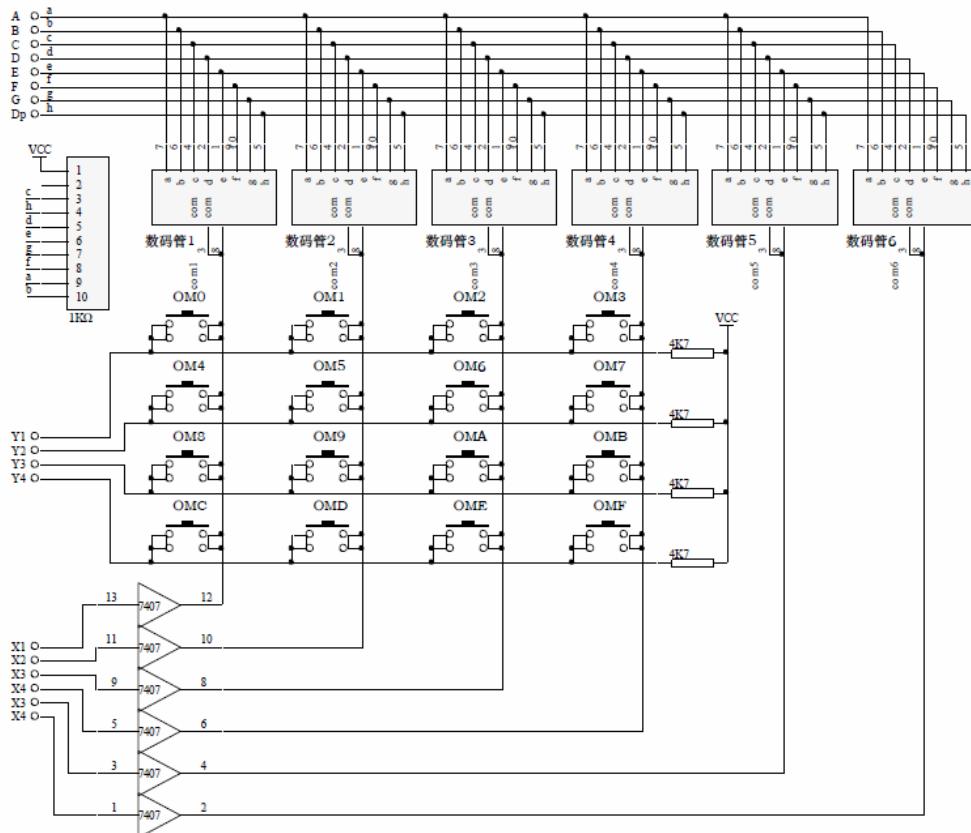


图 1-10 键盘扫描及数码管显示实验单元电路

1.4.10 点阵 LED 显示单元

点阵单元由 4 块 8×8 LED 器件组成, SM 16206 是 LED 恒流驱动芯片, 内建 CMOS 移位寄存器与锁存功能, 可以将串行的输入数据转换成并行输出数据格式。SM5166PC 是高集成度 LED 驱动控制芯片, 内置 3-8 译码器, 提供 8 通道的输出电流驱动。

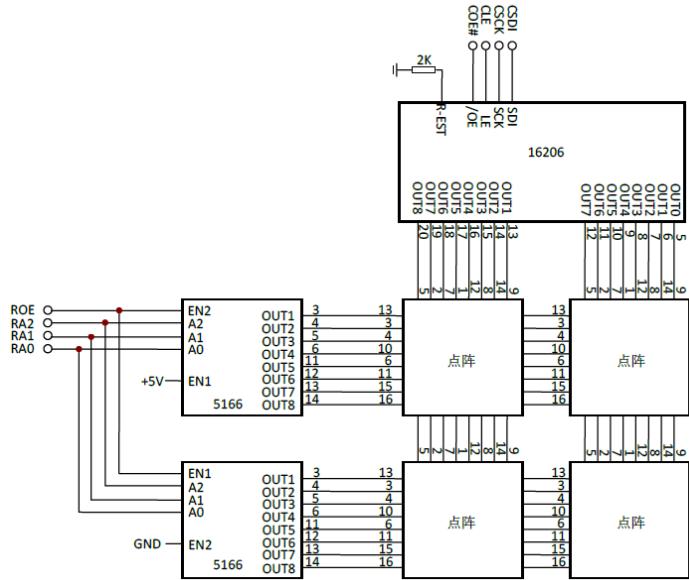


图 1-11 点阵 LED 显示单元电路

1.4.11 驱动电路单元

驱动电路如图 1-12 所示。UNL2003 为驱动接口芯片，由该芯片组成驱动电路，输入端 N 经过一个反相器连接到 2003 的输入端，其它四路 A、B、C、D 不经过反相器直接与 2003 相连。

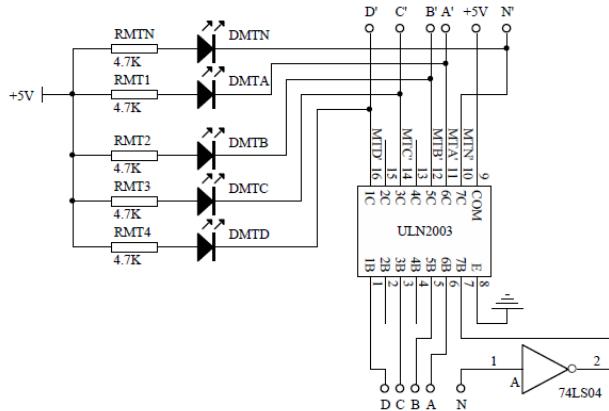


图 1-12 驱动电路实验单元

1.4.12 步进电机与直流电机单元

步进电机为四相八拍电机，如图 1-13 所示。直流电机单元由 DC5V 直流电机及霍尔器件组成，如图 1-14 所示。

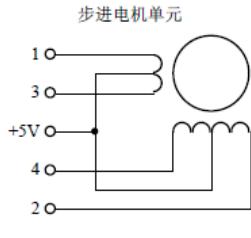


图 1-13 步进电机

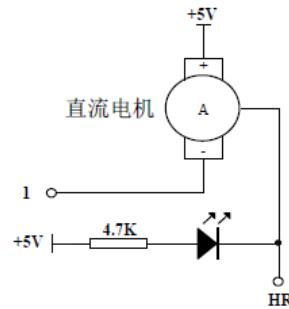


图 1-14 直流电机

1.4.13 电子发声单元与温度控制单元

电子发声单元由放大电路与扬声器组成，如图 1-15 所示。温度控制单元主要由 PTC 陶瓷加热片、热敏电阻组成，如图 1-16 所示。

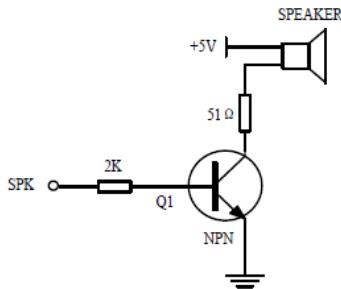


图 1-15 电子发声单元

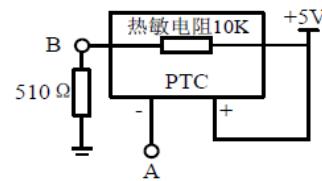


图 1-16 温度控制单元

1.4.14 单次脉冲单元

该单元提供两组消抖动单次脉冲，分别为 KK1-、KK1+、KK2-、KK2+，如图 1-17 所示。‘-’表示按下按键为低电平，‘+’表示按下按键为高电平。

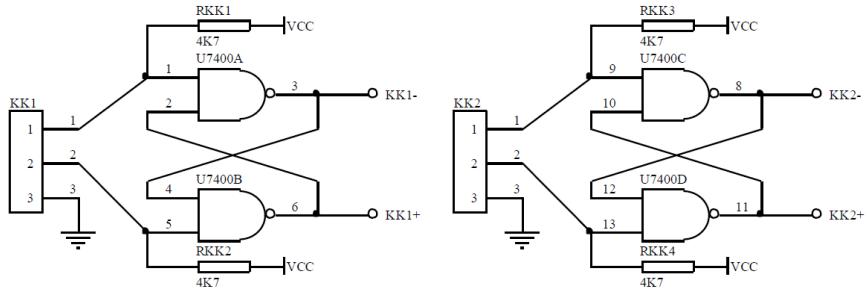


图 1-17 单次脉冲单元电路

1.4.15 逻辑开关及 LED 显示单元

逻辑开关及 LED 显示单元由 16 组开关及 16 个 LED 组成。16 组开关未经过消抖动，16 个 LED 灯显示逻辑电平高低，为正逻辑，输入高电平 LED 点亮。如图 1-18 所示。

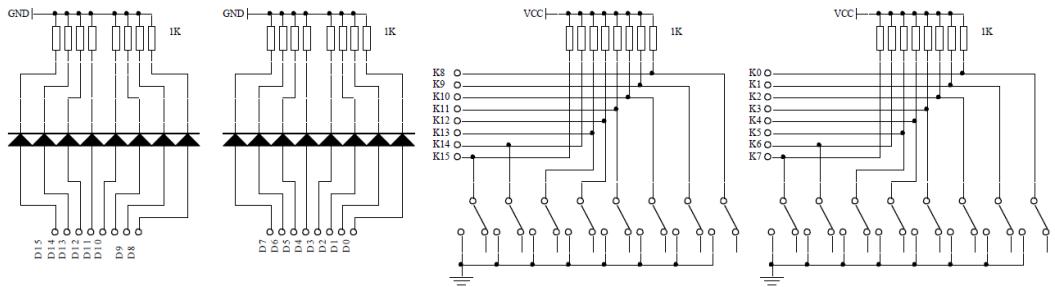


图 1-18 逻辑开关及 LED 显示单元电路

1.4.16 地址译码单元

该单元提供一片开放的译码器 74LS138，用于学习地址译码方法。其线路连接如图 1-19 所示。

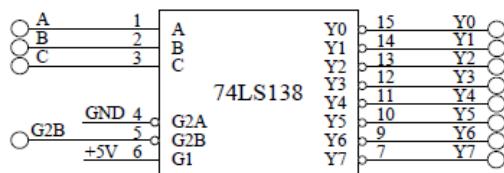


图 1-19 地址译码单元电路

1.4.17 16 位 I/O 接口单元

该单元通过 2 片三态缓冲器和 2 片锁存器组成 16 位的 I/O 接口，并根据 16 位总线时序设计了译码电路，可以 8/16 位不同字节宽度来访问该接口。用于学习 8 位和 16 位 I/O 接口设计及编程。其线路连接如图 1-20 所示。

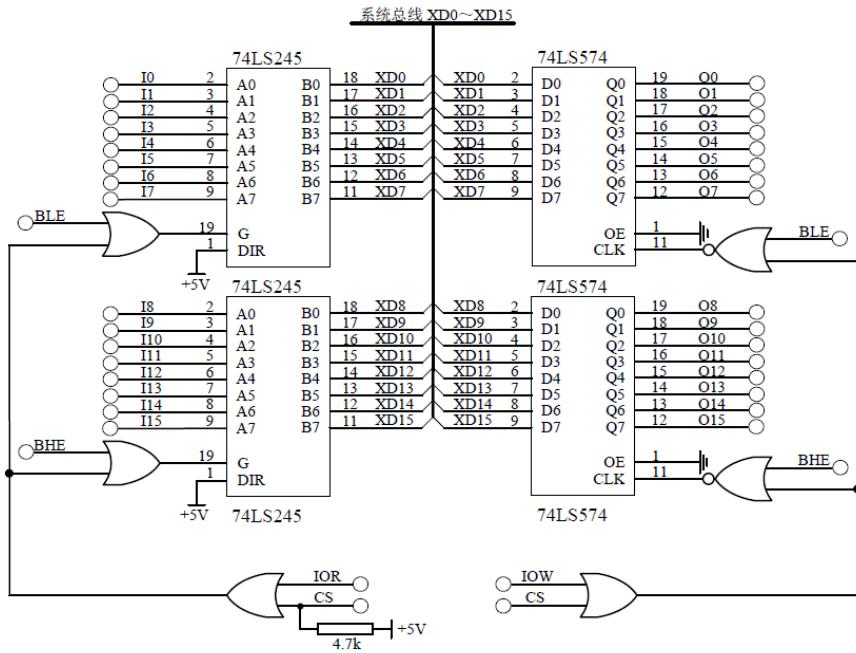


图 1-20 用两组 8 位 I/O 接口组成的 16 位 I/O 接口电路

1.4.18 转换单元

转换单元提供了排线和圆锥孔相互转接及扩展的单元。

1.4.19 扩展单元

扩展单元提供 2 组 40 线通用集成电路扩展单元和一个扩展模块总线单元。扩展模块总线插座的信号定义如图 1-21 所示。

其中，N11、N12、N13、N14、N15、N16、N17、N18 这 8 个引脚已联接到对应于它下方 40 线通用集成电路扩展单元的 11、12、13、14、15、16、17、18 脚上，可根据需要定义。其余的引脚与 X86 系统总线单元各引脚相对应。

扩展区右上角 JMC51 的 4 脚座，是在选配 TD-51 单片机开发板时的安装接口，从第 1 脚开始，其信号依次为 RXD、TXD、GND、+5V。

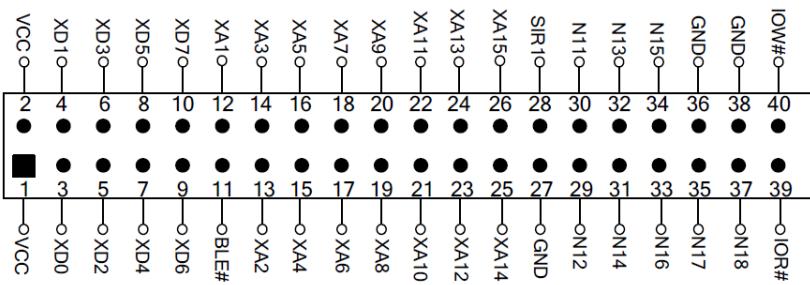


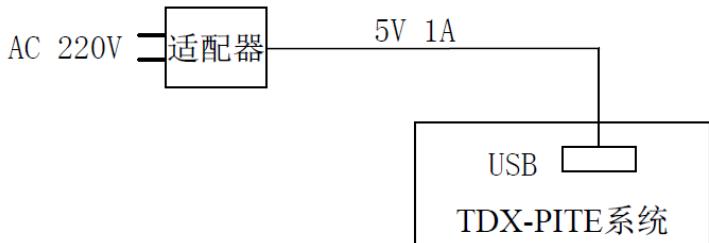
图 1-21 扩展模块总线 EX_BUS 单元引脚示意图

1.4.20 系统总线单元

系统总线单元各引脚如前表 1-2 所示。其中 CPU 选择开关分为 51 和 386 两个档位。开关打到 386 档，表明 386CPU 与设备箱体上的 USB 口相连可与 PC 机通讯；开关打到 51 档，表明选配的 TD-51 单片机开发板与设备箱体上的或 USB 口相连可与 PC 机通讯。

1.5 系统电源

TDX-PITE 系统所配+5V、1A 适配器，通过 USB 电缆向实验系统提供+5V 电源，如下图所示。



第二章 TDX-PITE 系统软件概述

TDX-PTE 联机软件是为 TD 系列微机系统与接口技术教学实验系统配套的集成开发调试软件。该软件具有汇编语言和 C 语言源语言级调试跟踪界面，有别于传统的 DEBUG 调试，操作更简单，视觉效果更直接。如果用户习惯于 DEBUG 调试，点击输出区的调试标签亦可对源程序进行 DEBUG 调试。

用户可以选择是否监控堆栈，也可以通过“变量监控”对话框选择监控哪些全局变量；寄存器变量/堆栈区能够对下位机的寄存器、变量和堆栈实现实时监视，使寄存器/变量/堆栈区的数据永远是下位机中数据的真实反映；在寄存器/变量堆栈区改写寄存器和变量的值，即可改变下位机中寄存器和变量的值。

TDX-PTE 联机软件也集成了一个“专用图形显示”、“示波器”虚拟仪器，可用于几个特殊实验波形的观测。

TDX-PITE 联机软件具有特点如下：

- (1) 支持汇编语言和 C 语言两种编译环境；
- (2) 具有 16 位寄存器和 32 位寄存器状态切换；
- (3) 高度可视化的源语言级调试跟踪界面；
- (4) 实时监视寄存器，能够即时对寄存器中的值进行修改；
- (5) 可以选择要监视的全局变量，进行实时监视和即时修改；
- (6) 可以选择是否要实时监视堆栈寄存器；
- (7) 集成虚拟仪器：专用图形显示、示波器窗口；
- (8) 独特的时序测量窗口，准确观测各总线信号的时序状态和关系。

2.1 TDX-PTE 联机软件使用说明

2.1.1 主窗口界面

主界面窗口主要分为四部分：程序编辑区、寄存器/变量/堆栈区、输出区和，如图 2-1 所示。

1. 程序编辑区

位于界面右上部，用户可在程序编辑区用“新建”命令创建一个新文档或用“打开”命令打开一个已存在的文档，在文档中用户可编辑程序。在程序编辑区可以同时打开多个文档，点击文档标签可激活任一文档。编译、链接、加载以及调试命令只针对当前活动文档。

2. 寄存器/变量/堆栈区

位于界面左上部，包括三个部分：寄存器区、变量区和堆栈区。寄存器区和变量区用于实时监视寄存器和变量，也能够修改寄存器和变量的值，堆栈区主要用于实时监视堆栈寄存器。点击寄存器区标签可观察和修改寄存器，点击变量区标签可观察和修改变量，点击堆栈区标签可观察堆栈寄存器。执行相关操作后寄存器、变量和堆栈寄存器的值自动刷新，如果寄存器区刷新后的值和刷新前的值相同，则显示为蓝色，如果不同，则显示为红色。变量区和堆栈区刷新前和刷新后都为蓝色。想修改寄存器或变量，用鼠标左键单击某寄存器或变量，当寄存器或变量值的区域变为蓝色后输入修改后的值，然后回车。如果修改的是寄存器，输入的值的长度与寄存器的长度相同，回车后修改正确。如果输入的值的长度和寄存器的长度不相同，寄存器的值没有被修改，保持原来的值且可以继续修改该寄存器的值。如果修改的是变量，输入的值的长度与变量的长度相同，回车后修改正确。如果输入的值的长度小于变量的长度，回车后输入值的前面自动添零；如果输入值的长度大于变量的长度，回车后输入值的后面的多余部分自动删除。堆栈区只用于监视，不能被修改。

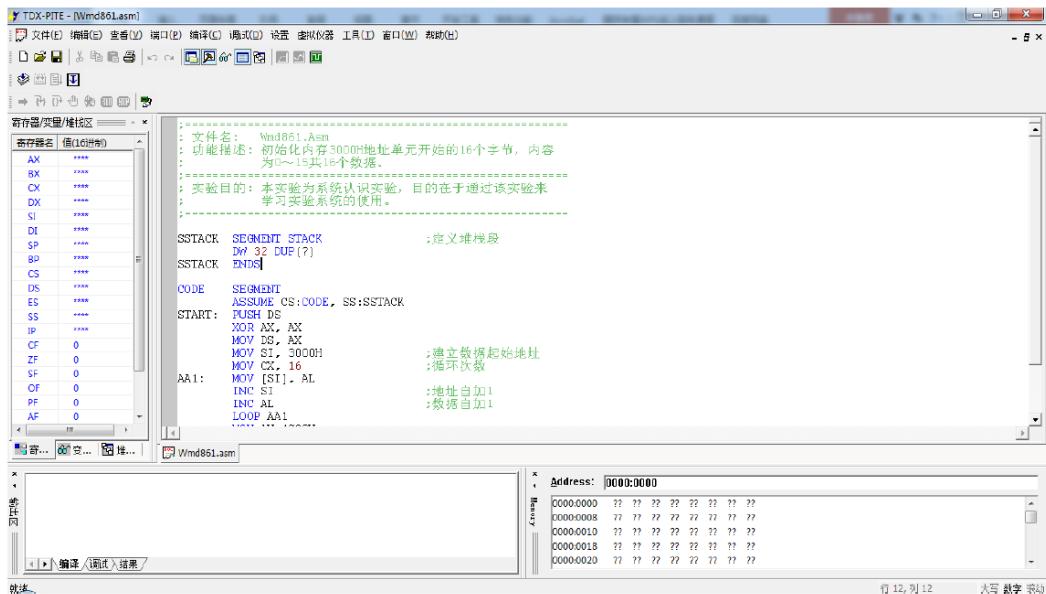


图 2-1 TDX-PTE 主窗口界面

3. 输出区

位于界面下部，输出区包含三个部分：编译区、调试区和结果区。编译区显示编译和链接的结果，如果编译时有错误或警告，双击错误或警告信息，错误标识符会指示到相应的有错误或警告的行。调试区主要用于 DEBUG 调试。结果区主要用于显示程序加载结果、程序运行输出结果和复位结果。结果区中有“加载中，请稍候....”的字样表示联机软件正在把可执行文件加载到下位机中，用户此时应等待直到加载完成，“加载成

功”表示加载完成且成功，“加载失败”表示加载完成且失败，此时应重新加载。程序中用“INT 21H”输出的结果亦显示在此区。软复位或硬件复位成功后，结果区显示“复位成功！”。如果复位不成功则不显示此句。

4. Memory 区

位于界面右下部，Memory 区包含两个部分：起始地址和数据区。在起始地址填写需要查询或修改的地址，数据区左侧表示对应的存储地址，右侧表示地址中存放的数据，可点击修改对应地址中的数据。

2.1.2 菜单功能

1. 文件菜单项

文件菜单如图 2-2 所示。

(1) 新建(N): 用此命令在 TDX-PITE 中建立一个新文档。

(2) 打开(O): 用此命令在窗口中打开一个现存的文档。可同时打开多个文档，点击某文档的标签可激活此文档。可用窗口菜单在多个打开的文档中切换。

(3) 关闭(C): 用此命令来关闭当前活动文档。TDX-PITE 会建议在关闭文档之前保存对文档所做的改动。如果没有保存而关闭了一个文档，将会失去自从最后一次保存以来所做的所有改动。在关闭一无标题的文档之前，TDX-PTE 会显示另存为对话框，建议命名和保存文档。

(4) 保存(S): 用此命令将当前活动文档保存到当前的文件名和目录下。当第一次保存文档时，TDX-PITE 显示另存为对话框以便命名文档。

(5) 另存为(A): 用此命令来保存并命名活动文档。TDX-PITE 会显示另存为对话框以便命名文档。

(6) 打印(P): 用此命令来打印一个文档。

(7) 打印预览(V): 用此命令按要打印的格式显示活动文档。当选择此命令时，主窗口就会被一个打印预览窗口所取代。这个窗口可以按文档被打印时的格式显示一页或两页。

(8) 打印设置(R): 用此命令来选择连接的打印机及其设置。

(9) 最近浏览文件: 通过此列表，直接打开最近打开过的文件，共四个。

(10) 退出(X): 用此命令来结束 TDX-PITE 的运行阶段。TDX-PITE 会提示保存尚未保存的改动。

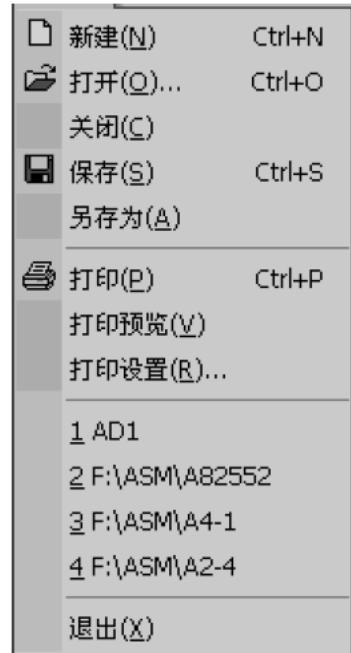


图 2-2 文件菜单

2. 编辑菜单项

编辑菜单如图 2-3 所示。

(1) 撤消: 可用此命令来撤消上一步操作。如果无法撤消上一步操作, 菜单上的撤消命令会变灰。

(2) 重复: 可用此命令来恢复撤消的编辑操作。如果无法恢复撤消的编辑操作, 菜单上的重复命令会变灰。

(3) 剪切(T): 用此命令将当前被选取的数据从文档中删除并放置于剪贴板上。当前没有数据被选取时, 则此命令不可用。把数据剪切到剪贴板上将取代存放在那里的内容。

(4) 复制(C): 用此命令将被选取的数据复制到剪贴板上。当前无数据被选取时, 此命令不可用。把数据复制到剪贴板上将取代存放在那里的内容。

(5) 粘贴(P): 用此命令将剪贴板上内容的一个副本插入到插入点处。如剪贴板是空的, 则此命令不可用。

(6) 查找: 点击此命令将弹出查找对话框, 如图 2-4 所示, 用于查找指定字符串。



图 2-3 编辑菜单

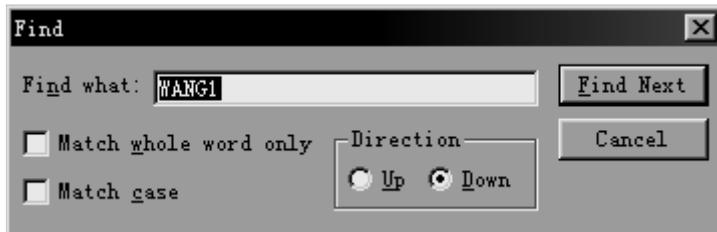


图 2-4 查找对话框

Find what 编辑框: 写入要查找的字符串。

Match whole word only 复选框: 是否全字匹配。如果不选中此复选框, 找到的字符串的长度有可能大于想要查找的字符串。例如, 想要查找字符串“WANG1”, 可能会找到字符串“WANG10”。

Match case 复选框: 是否辨认大小写。如果不选中此复选框, 找到的字符串中字符的大小写可能与想要查找的字符串有差别。例如想要查找字符串“WANG1”, 可能会找到字符串“Wang1”。

Up 单选按钮: 从下向上查找。

Down 单选按钮: 从上向下查找。

Find Next 按钮: 查找下一个字符串。如果是第一次查找则从当前光标处开始向下或向上开始查找; 如果不是第一次查找, 则从上一次找到的位置向下或向上开始查找。

Cancel 按钮: 取消查找对话框。

(7) 替换：点击此命令将弹出替换对话框，如图 2-5 所示，找到某一字符串，并用指定字符串替换之。

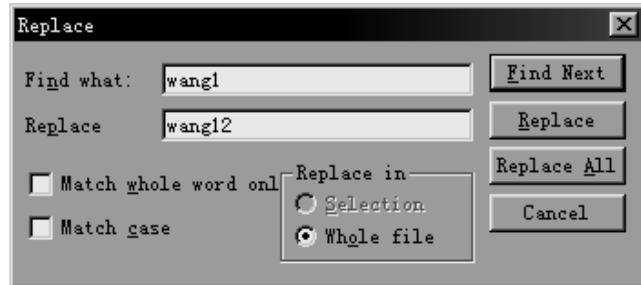


图 2-5 替换对话框

Replace 编辑框：替换后的字符串。

Selection 单选按钮：如果文档中有选中部分，此按钮使能，选中此按钮则从选中部分查找和替换。

Whole file 单选按钮：从整个文档中查找和替换。

Find Next 按钮：查找下一个字符串。如果是第一次查找，从当前光标位置开始查找；如果不是第一次查找，则从上一次找到的位置开始查找。

Replace 按钮：替换一个字符串。如果当前已经找到某一字符串，用指定字符串替换它，并找到下一个字符串；如果还没有找到某一字符串，不进行替换并找到字符串。

Replace All 按钮：用指定字符串替换全部能够找到的字符串。

Cancel 按钮：取消替换对话框。

3. 查看菜单项

查看菜单如图 2-6 所示。

(1) 工具栏(T)：显示或隐藏工具栏。

a. 标准工具栏：用此命令可显示和隐藏标准工具栏。标准工具栏包括了 TDX-PITE 中一些最普通命令的按钮，如文件打开。在工具栏被显示时，一个打勾记号出现在该菜单项目的旁边。

b. 变量区(W)：用此命令可显示和隐藏寄存器/变量/堆栈区。

c. 输出区(O)：用此命令可显示和隐藏输出区。

d. 编译工具栏(B)：用此命令可显示和隐藏编译工具栏。

e. 调试工具栏(D)：用此命令可显示和隐藏调试工具栏。

f. 自定义(C)：见自定义功能。

(2) 状态栏(S)：显示或隐藏状态栏。



图 2-6 查看菜单

4. 端口菜单项

端口菜单如图 2-7 所示。

(1) 端口选择：此命令用来选择 COM1、COM2...进行联机通讯，该命令会对 COM 口进行初始化操作。

(2) 端口测试：此命令用来对当前选择的 COM 口进行联机通讯测试，并报告测试结果。



图 2-7 端口菜单

5. 编译菜单项

编译菜单如图 2-8 所示。

(1) 编译(C)：编译当前活动文档中的源程序，在源文件目录下生成目标文件。如果有错误或警告生成，则在输出区显示错误或警告信息，双击错误或警告信息，可定位到有错误或警告的行，修改有错误或警告的行后应重新“编译”。如果编译没有错误生成（即使有警告生成）使能“链接”菜单项和工具栏中的“链接”按钮，以便进行链接。编译时自动保存源文件中所做的修改。

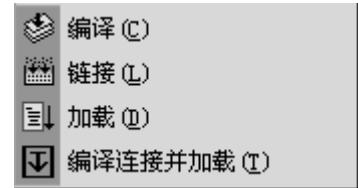


图 2-8 编译菜单

(2) 链接(L)：链接编译生成的目标文件，在源文件目录下生成可执行文件。如果有错误或警告生成，则在输出区显示错误或警告信息，查看错误或警告信息修改源程序，修改后应重新“编译”和“链接”。如果链接没有错误生成（即使有警告生成），使能“加载”菜单项和工具栏中的“加载”按钮，以便进行加载。

(3) 加载(D)：把链接生成的可执行文件加载到下位机。在加载过程中输出区有“加载中，请稍候...”的字样，用户此时应该等待直到加载完成。加载完成后，如果加载成功，输出区显示“加载成功！”，使能“调试”菜单中的菜单项和调试工具栏中的按钮，此时 CS 和 IP 指向程序的开始执行并在此行设置执行标记。如果加载失败，输出区显示“加载失败！”，不使能“调试”菜单中的菜单项和调试工具栏中的按钮，此时应重新进行“加载”。

(4) 编译链接并加载(T)：依次执行编译、链接和加载。

6. 设置菜单

设置菜单如图 2-9 所示。

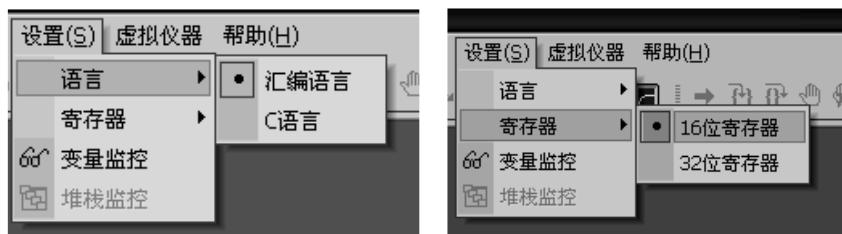


图 2-9 设置菜单

(1) 语言：设置语言环境

汇编语言：设置编译环境为汇编语言环境。此时可编辑、编译和链接 BM-PC 汇编语言源程序。

C 语言：设置编译环境为 C 语言环境。此时可编辑、编译和链接 C 语言源程序。由于监控目前不支持浮点运算，故 C 语言程序中不应该出现浮点运算，如果 C 语言程序中出现浮点运算，链接时会出现错误。

(2) 寄存器：设置寄存器格式

16 位寄存器：设置成 16 位寄存器，可观察到 16 位寄存器的变化。

32 位寄存器：设置成 32 位寄存器，可观察到 32 位寄存器的变化。

(3) 变量监控：加载成功后才可用此按钮。系统只能监视全局变量。在汇编语言源文件中，数据段定义的变量并不是全局变量，因此数据段定义的变量并不出现在图 2-10 所示的对话框的左边列表，要想监视这些变量，必须使它们成为全局变量。使一个变量成为全局变量的方法是用关键字 PUBLIC 在源程序的最前面声明之。



图 2-10 变量监控

左边的列表框为程序中的全局变量，系统只能监视全局变量，右边的列表框为系统正在监视的全局变量。选择需要监视的变量，点击“加入监视”按钮，以实现对该变量的监视。如果想从变量区去掉某一正在监视的变量，选中变量后点击“停止监视”按钮将其从列表中删除。

在汇编语言源文件中，数据段定义的变量并不是全局变量，因此数据段定义的变量并不出现在图 2-10 所示的对话框的左边列表，要想监视这些变量，必须使它们成为全局变量。使一个变量成为全局变量的方法是用关键字 PUBLIC 在源程序的最前面声明之，格式是：PUBLIC symbol[,...]，范例如下：

```
PUBLIC mus_time  
PUBLIC mus_freq  
  
DATA1 SEGMENT  
mus_time DB 01H  
mus_freq DW 1234H  
DATA1 ENDS
```

数据段 DATA1 中的数据 mus_time, mus_freq 经过 PUBLIC 声明后成为全局变量，编译、链接、加载完成后，可对这两个变量进行监控。

在 C 语言源文件中，函数内部定义的变量不是全局变量，函数外面定义的变量才是全局变量，因此系统只能监视函数外面定义的变量。要想监视某一变量，应该把它定义在函数的外面。

(4) 堆栈监控

堆栈监控如图 2-11 所示。选择“不监控堆栈”单选按钮，确定后不监视堆栈寄存器；选择“监控堆栈”单选按钮，确定后监视堆栈寄存器。默认选项为“不监控堆栈”。

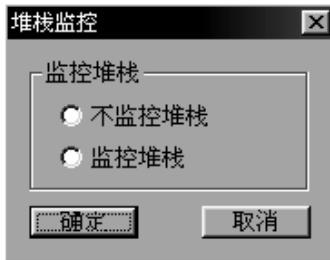


图 2-11 堆栈监控

7. 调试菜单项

调试菜单如图 2-12 所示。



图 2-12 调试菜单项

(1) 设置断点/删除断点(B): 当前光标所在的行为当前行。如果当前行无断点则在当前行设置断点；如果当前行有断点则删除当前行的断点。

```
add al,01h  
call WW31  
mov al,05h  
add al,11h
```

并不是源程序的所有行都可以设置断点，如伪操作行和空行不能设置断点。源程序设置

的断点数不能超过 8 个。

(2) 清除所有断点(D): 清除源程序中设置的所有断点。只有当设置的断点数大于零时，该菜单才使能。

(3) 设置起点(J): 当前光标所在的行为当前行，此命令把当前行设置为程序的起点，即程序从此行开始运行，寄存器区的 CS 和 IP 的值刷新后指向此行。并不是源程序的所有行都可以设置起点，如伪操作行和空行不能设置起点。

```
CODE SEGMENT  
ASSUME CS:CODE,DS:DAT  
→ START: mov ax,DAT  
          mov ds,ax  
          call far ptr ww1  
          add al,01h  
          add al,01h
```

(4) 单步(S): 点击此命令使程序执行一条语句，如果是函数则进入函数内部，执行后刷新所有的变量和寄存器的值。并不是所有的语句行都适用单步，如系统调用语句不应该使用单步，而应该用跳过命令跳过该语句行。

(5) 跳过(O): 点击此命令使程序执行一个函数，执行后刷新所有变量和寄存器的值。只有当当前执行行为函数调用或系统调用，才用此命令。如果函数内部有断点，点击“跳过”指令后，程序会停在函数内部有断点的行。

(6) 运行/运行到断点: 从当前执行行开始向后运行。如果没有断点，则运行直到程序结束；如果有断点，则运行到断点后停止。运行到断点后如果再次点击此菜单，则从当前断点位置继续执行，直到再次遇到断点或程序结束。

(7) 停止: 发送此命令使程序停止运行，程序停止后刷新所有寄存器和变量。

(8) 固化程序: 将实验程序固化到系统存储器 FLASH 中，以实现程序的脱机运行。此命令只有在程序正常加载后执行。

8. 虚拟仪器菜单项

具体各项说明详见专用图形显示介绍。



9. 窗口菜单项

如图 2-13 所示，窗口菜单提供了以下命令，这些命令使您能在应用程序窗口中安排多个文档的多个视图。

(1) 新建窗口(N): 用此命令来打开一个具有与活动的窗口相同内容的新窗口。可同时打开数个文档窗口以显示文档的不同部分或视图。如果对一个窗口的内容做了改动，所有其它包含同一文档的窗口也会反映出这些改动。当打开一个新的窗口，这个新窗口就成了活动的窗口并显示于所有其它打开窗口之上。

(2) 层叠(C): 用此命令按相互重叠形式来安排多个打开的窗口。

(3) 平铺(T): 用此命令按互不重叠形式来安排多个打开的窗口。

(4) 排列图标(A): 用此命令在主窗口的底部安排被最小化的窗口的图标。如果在



图 2-13 窗口菜单

主窗口的底部有一个打开的窗口，则有可能会看不见某些或全部图标，因为它们在这个文档窗口的下面。

(5) 窗口 1,2,...: TDX-PITE 在窗口菜单的底部显示出当前打开的文档窗口的清单。有一个打勾记号出现在活动的窗口的文档名前。从该清单中挑选一个文档可使其窗口成为活动窗口。

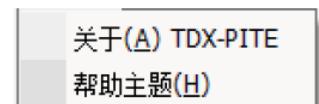
10. 帮助菜单项

帮助菜单提供使用这个应用程序的帮助。

(1) 帮助主题(H): 用此命令来显示帮助的开场屏幕。

从开场屏幕可跳到关于使用 TDX-PITE 的一些指令以及各种不同类型参考资料。

(2) 关于(A) TDX-PITE: 用此命令来显示 TDX-PITE 版本的版权通告和版本号码。



2.1.3 工具栏功能介绍

1. 标准工具栏

标准工具栏共有十七个按钮，如下图所示。



- (1)  按钮：用此按钮在TDX-PITE 中建立一个新文档。
- (2)  按钮：用此命令在一个新的窗口中打开一个现存的文档。
- (3)  按钮：用此命令将当前活动文档保存到其当前的文件名和目录下。
- (4)  按钮：用此命令将当前被选取的数据从文档中删除并放置于剪贴板上。
- (5)  按钮：用此命令将被选取的数据复制到剪切板上。
- (6)  按钮：用此命令将剪贴板上内容的一个副本插入到插入点处。
- (7)  按钮：用此命令来打印一个文档。
- (8)  按钮：用此命令来撤消上一步编辑操作。
- (9)  按钮：用此命令来恢复撤消的编辑操作。
- (10)  按钮：用此按钮可显示和隐藏变量和寄存器区。
- (11)  按钮：用此按钮可显示和隐藏输出区。
- (12)  按钮：加载成功后才可用此按钮。点击此按钮，可进行全局变量监视。
- (13)  按钮：用此按钮可显示和隐藏Memory 区。
- (14)  按钮：堆栈监控按钮，点击此按钮将弹出堆栈监控对话框。
- (15)  按钮：用来启动专用图形显示。

- (16)  按钮：用来启动示波器功能。
- (17)  按钮：用来启动时序观测窗功能。

2. 编译工具栏

编译工具栏共有四个按钮，如下图所示。



- (1)  编译：编译当前活动文档中的源程序，在源文件目录下生成目标文件。
- (2)  链接：链接编译生成的目标文件，在源文件目录下生成可执行文件。
- (3)  加载：把链接生成的可执行文件加载到下位机。
- (4)  编译链接并加载：依次执行编译、链接和加载。

3. 调试工具栏

调试工具栏共有七个按钮，如下图所示。



- (1)  设置起点：当前光标所在的行为当前行，此命令把当前行设置为程序的起点，即程序从此行开始运行，寄存器区的 CS 和 IP 的值刷新后指向此行。
- (2)  单步：点击此命令使程序执行一条语句。
- (3)  跳过：点击此命令使程序执行一个函数，执行后刷新所有变量和寄存器的值。
- (4)  设置断点/删除断点：为光标所在行设置断点或删除当前行的已有断点。源程序设置的断点数不能超过 8 个。
- (5)  清除所有断点：清除源程序中设置的所有断点。
- (6)  运行到断点/运行：从当前执行行开始向后运行，如果没有断点，则运行直到程序结束；如果有断点，则运行到断点后停止。运行到断点后再次点击此按钮，则程序从当前断点位置继续执行，直到再次遇到断点或程序结束。
- (7)  停止：发送此命令使程序停止运行，程序停止后刷新所有寄存器和变量的值。

2.1.4 专用图形显示

主要用于观察“直流电机闭环调速实验”及“温度单元或电烤箱闭环温度控制实验”的响应曲线，本界面可以观察系统的给定值、反馈值及控制量之间的响应曲线关系。如图 2-14 所示。

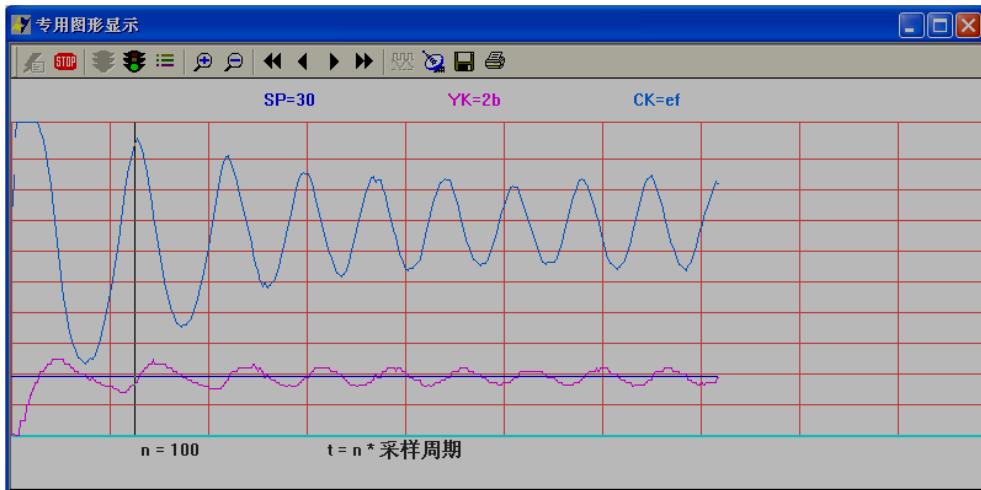


图 2-14 专用图形显示界面

工具栏功能简介：

- (1) 按钮：启动并运行程序。运行加载在下位机中的程序。“SP=” 后显示的值是当前时刻系统的给定值，“YK=” 后显示的值是当前系统的反馈值，“CK=” 后显示的值是当前时刻系统控制量的值。
- (2) 按钮：停止程序运行。使下位机中运行的程序停止。
- (3) 按钮：暂停程序运行。在运行状态下使波形暂停显示并出现游标。
- (4) 按钮：退出暂停状态，使波形继续显示，游标消失。
- (5) 按钮：显示选择按钮，可选择性的选择要显示的波形。
- (6) 按钮：放大波形。
- (7) 按钮：缩小波形。
- (8) 按钮：快速左移游标。在暂停状态下，使游标快速向左移动。“SP=” 后显示的值是游标所在时刻系统的给定值，“YK=” 后显示的值是游标所在时刻系统的反馈值，“CK=” 后显示的值是游标所在时刻系统控制量的值。
- (9) 按钮：左移游标。在暂停状态下，使游标向左移动。“SP=” 后显示的值是游标所在时刻系统的给定值，“YK=” 后显示的值是游标所在时刻系统的反馈值，“CK=” 后显示的值是游标所在时刻系统控制量的值。
- (10) 按钮：右移游标。在暂停状态下，使游标向右移动。“SP=” 后显示的值是游标所在时刻系统的给定值，“YK=” 后显示的值是游标所在时刻系统的反馈值，“CK=” 后显示的值是游标所在时刻系统控制量的值。
- (11) 按钮：快速右移游标。在暂停状态下，使游标快速向右移动。“SP=” 后

显示的值是游标所在时刻系统的给定值，“YK=”后显示的值是游标所在时刻系统的反馈值，“CK=”后显示的值是游标所在时刻系统控制量的值。

- (12) 按钮：记录波形。
- (13) 按钮：显示保存到图一，图二和图三中的波形。
- (14) 按钮：打印当前屏幕上的波形。

2.1.5 示波器

利用本实验平台的 A/D 转换单元，可实现 20ms 采样周期的示波器功能，主要用于“8254 定时/计数器实验”、“D/A 转换实验”及“8251 串行接口实验”中波形的观察。实验时，需要把 A/D 单元接入系统，其接线方式如图 2-15 所示。

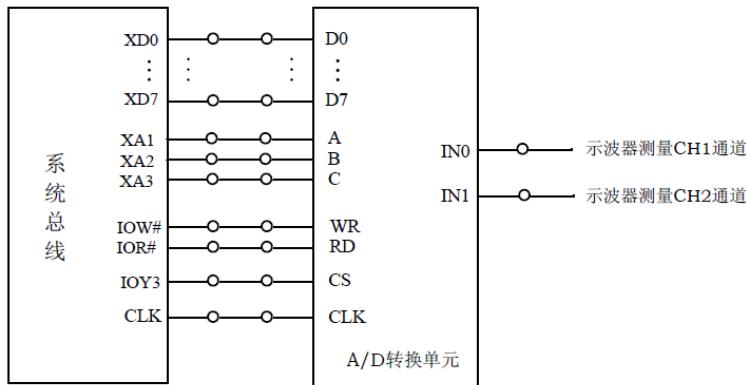


图 2-15 示波器接线方式

实验时示波器显示如图 2-16 所示。

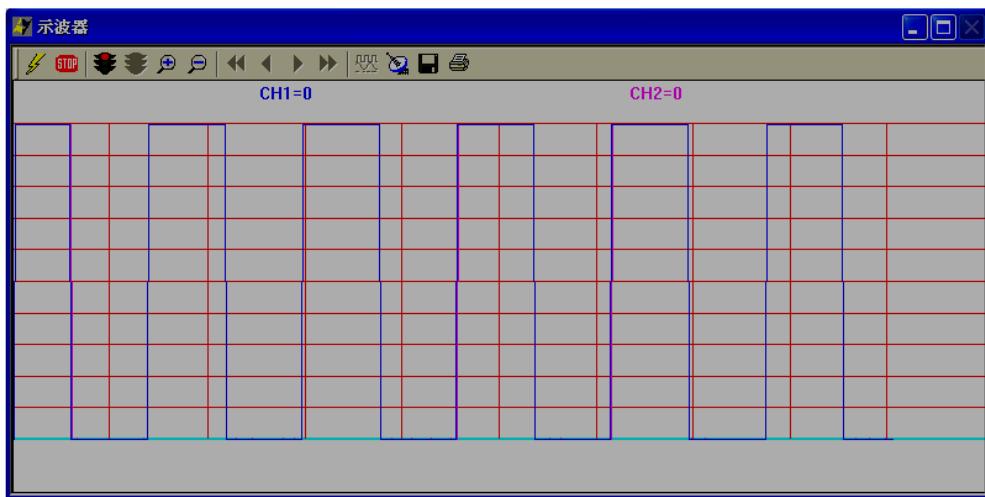


图 2-16 示波器显示界面

- (1)  按钮：启动示波器，“CH1=”，“CH2=”后分别显示游标当前位置的采样值。
- (2)  按钮：停止，使下位机中运行的程序停止。
- (3)  按钮：在运行状态下使能，使波形暂停显示并出现游标。
- (4)  按钮：在暂停状态下使能，使波形继续显示，游标消失。
- (5)  按钮：放大波形。
- (6)  按钮：缩小波形。
- (7)  按钮：在暂停状态下，使游标快速向左移动。“CH1=”、“CH2=”后分别显示游标当前位置的采样值。“T = xxx”表示游标所在位置的时刻与图形最左端时刻的差值。
- (8)  按钮：在暂停状态下，使游标向左缓慢移动。
- (9)  按钮：在暂停状态下，使游标向右缓慢移动。
- (10)  按钮：在暂停状态下，使游标快速向右移动。
- (11)  按钮：点击此按钮，出现如右图所示对话框。



选中“图一”单选按钮，点击确定，系统会把当前时刻的波形保存到图一中，共可保存三副图。图形只是保存于数据缓冲中，供图形比较时使用。

- (12)  按钮：显示保存到图一、图二和图三中的波形，此时可以对几幅图进行比较。
- (13)  按钮：以.bmp 格式保存当前屏幕上的波形到指定文件。
- (14)  按钮：打印当前屏幕上的波形。

2.1.6 时序观测

可对总线信号及各个指令操作的时序状态和时序关系进行准确的观测和分析，包含 I/O 操作、中断、DMA 等多种类型的观测。

在时序观测界面点击“右键”，弹出“选择观察信号”界面，可根据需要选择想要观测的信号或数据，点击确定。运行实验程序直至停止，观察各信号的时序图。如图 2-17 所示。

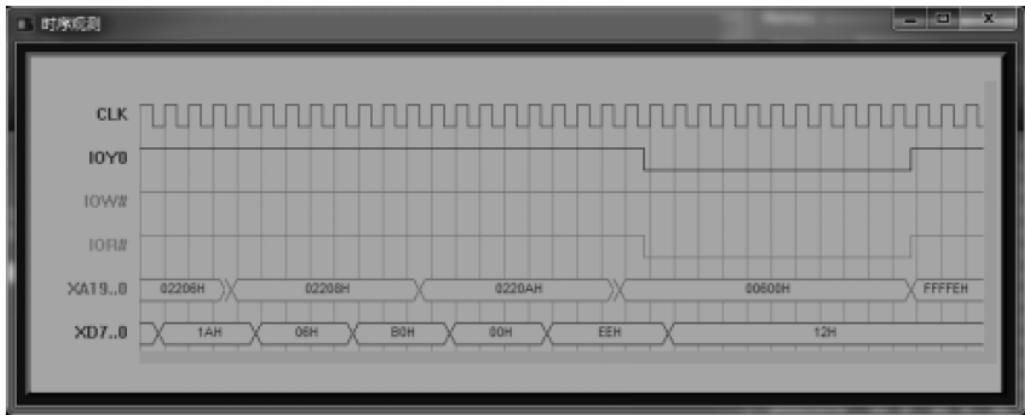


图 2-17 时序观测界面

2.1.7 自定义功能

点击“查看”菜单中“工具栏”项中的“自定义”菜单项，弹出“自定义”对话框，如图 2-18 所示。

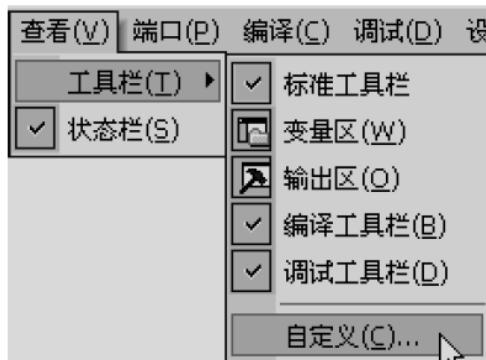


图 2-18 查看菜单-自定义菜单项

各属性页的功能如下：

1. 命令属性页

该属性页如图 2-19 所示。该属性页主要分成三块：类别、命令、说明。

- (1) 类别：列出了应用程序所有的菜单。
- (2) 命令：列出了选中的菜单中所对应的菜单命令。
- (3) 说明：如果选中了一个菜单命令，则在说明栏中注明该菜单命令的作用。



图 2-19 自定义菜单-命令

2. 工具栏属性页

这一属性页用来设置工具栏，如图 2-20 所示。系统提供了四个工具栏，可以在此控制其显示或隐藏。如果选中某一工具栏，按下“重新设置(R)”按钮恢复选中工具栏的默认设置；按下“重命名(m)”按钮可以修改选中工具栏的名称；按下“删除(D)”按钮删除选中工具栏；勾上右下角的“显示文本(S)”，使选中工具栏每一按钮下都显示该按钮的名称。点击“新建(N)”按钮新建一个工具栏。点击“全部重新设置(A)”按钮恢复全部工具栏的默认设置。



图 2-20 自定义菜单-工具栏

3. 工具属性页

该属性页用来设置“工具”菜单的菜单项，如图 2-21 所示。“工具”菜单本来为空，用户可向其中加入或删除菜单项。“菜单目录[M]:”列表框列出“工具”菜单中的现有菜单项。按钮为“工具”菜单新增一个菜单项；删除选中的菜单项；两个按钮调整选中菜单项的上下位置。选中某一菜单项，“命令[C]”编辑框中的内容表示

点击选中菜单项时，系统要执行的命令。如，点击“工具”菜单中的“MATLAB”菜单项，系统会执行“D:\MATLAB6pl\bin\win32\matlab.exe”命令，系统会为用户打开MATLAB。

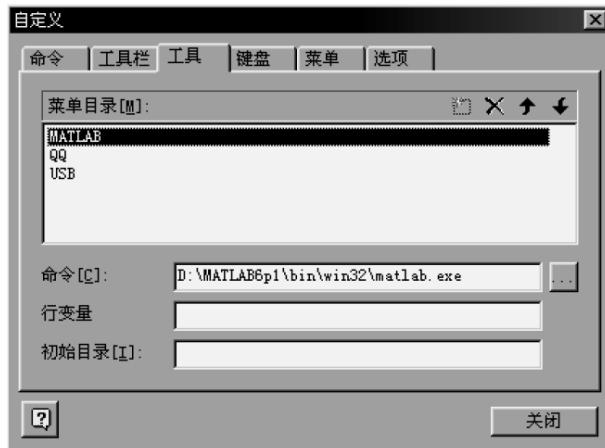


图 2-21 自定义菜单-工具

4. 键盘属性页

该属性页用来设置菜单命令的快捷键，如图 2-22 所示。方法是：在“类别”栏选择菜单，然后在“命令”栏选择需设置快捷键的菜单命令，此时说明栏会列出该菜单命令的作用，如果该菜单命令已设置了快捷键，则在“快捷键”栏会显示已设置的快捷键。如要设置新的快捷键，则可在“请按新快捷键”栏中键入新的快捷键，然后按下指定按钮即可。如要删除已有的快捷键，只要在“快捷键”栏中选中需删除的快捷键，然后按下删除按钮即可。



图 2-22 自定义菜单-键盘

5. 菜单属性页

在该属性页中，可以设置程序运行时主框架的菜单，但建议用默认值，也可以设置

菜单展开时的动画，以及是否产生菜单阴影。

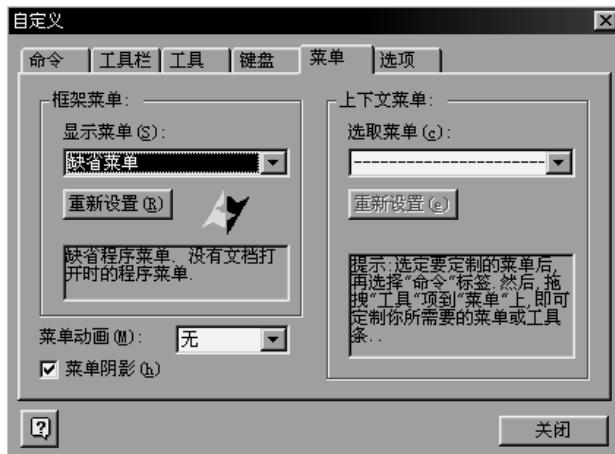


图 2-23 自定义菜单-菜单

6. 选项属性页

该属性页用来设置工具栏的一些特性，如图 2-24 所示。

如果勾上“显示关于工具栏的提示(T)”，使鼠标在某一按钮上停留片刻，则会显示该按钮的作用提示条。如果勾上“在屏幕提示中显示快捷方式(K)”，则在显示按钮作用提示条时还会显示该按钮的快捷键。

大图标：使工具栏按钮显示为大图标。

在短暂延迟后显示全部菜单项(u)：点击某一菜单，在短暂延迟后显示全部菜单项。

重新配置用户设置(R)：恢复默认设置。

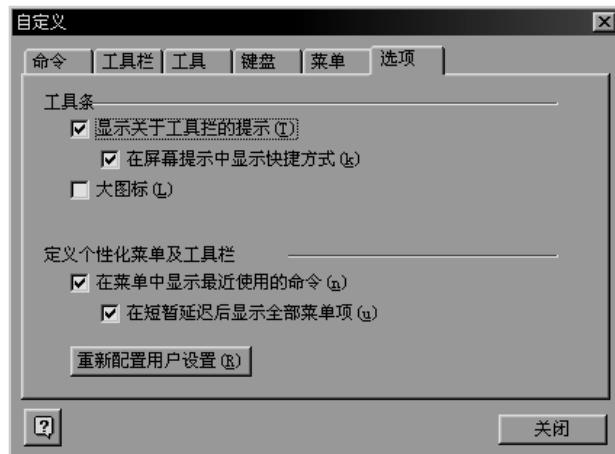


图 2-24 自定义菜单-选项

2.1.8 右键菜单功能

如果用户在程序编辑区点击右键，出现编辑菜单；如果在非客户区点击右键，出现工具栏菜单。分别介绍如下：

1. 编辑菜单

编辑菜单提供了“剪切”、“复制”、“粘贴”命令。

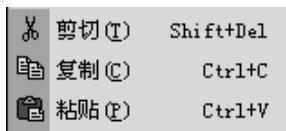


图 2-25 编辑菜单

2. 工具栏菜单

工具栏菜单与查看菜单中的工具栏的下拉菜单内容相同，功能亦相同。



图 2-26 工具栏菜单

2.1.9 Debug 调试命令

TDX-PITE 软件输出区集成有 Debug 调试，点击调试标签，进入 Debug 状态，会出现命令提示符“>”，主要命令如下：

A 进入小汇编

格式：A[段址:][偏移量]⁴

A 段址:偏移量⁴——从段址:偏移量构成的实际地址单元起填充汇编程序的目标代码；

A 偏移量⁴——从默认的段址与给定的偏移量构成的实际地址单元起填充汇编程序目标代码；

A⁴——从默认段址:默认偏移量构成的实际地址单元起填充汇编程序的目标代码。

输入上述命令后，屏幕显示地址信息，即可输入源程序。若直接回车，则退出命令。汇编程序输入时，数据一律为十六进制数，且省略 H 后缀。[m]类操作一定要在[]之前标注 W（字）或 B（字节）。如：MOV B[2010], AX, MOV W[2010], AX。

例：在“>”提示符下键入“A 2000⁴”，此时默认的段址 CS 为 0000，规定偏移量

IP 为 2000，屏幕显示与操作如表 2-1 所示。

表 2-1 小汇编操作示例

| 显示内容 | 键入内容 |
|-----------|----------------|
| 0000:2000 | MOV AX, 1234 ↵ |
| 0000:2003 | INC AX ↵ |
| 0000:2004 | DEC AX ↵ |
| 0000:2005 | JMP 2000 ↵ |
| 0000:2007 | ↵ |

B 断点设置

在系统提示符下，键入 **B ↵**，系统提示[i]:，等待输入断点地址。输入断点地址后回车，系统继续提示[i+1]:。若直接键入回车，则结束该命令。系统允许设置最多 10 个断点，断点的清除只能是通过系统复位或重新上电来实现。**B** 命令示例如表 2-2 所示。

表 2-2 B 命令示例

| 显示内容 | 键入内容 |
|------|--------|
| > | B ↵ |
| [0]: | 2009 ↵ |
| [1]: | ↵ |

D 显示一段地址单元中的数据

格式：D[[段址:]起始地址, [尾地址]] ↵

D 命令执行后屏幕上显示一段地址单元中的数据，在显示过程中，可用 **Ctrl+S** 来暂停显示，用任意键继续；也可用 **Ctrl+C** 终止数据显示，返回监控状态。

E 编辑指定地址单元中的数据

格式：E[[段址:]偏移量] ↵

该命令执行后，则按字节显示或修改数据，可通过“空格”键进入下一高地址单元数据的修改，使用“—”键则进入下一低地址单元进行数据的修改，并可填入新的数据来修改地址单元的内容。若输入回车，则结束 **E** 命令。**E** 命令示例如表 2-3 所示。

表 2-3 E 命令示例

| 显示内容 | 键入内容 |
|---------------|---------|
| > | E3500 ↵ |
| 0000:3500 00_ | 05 空格 |
| 0000:3501 01_ | 空格 |
| 0000:3502 02_ | — |
| 0000:3501 01_ | ↵ |

G 运行程序

格式： G=[段址:]偏移量 ↵

GB=[段址:]偏移量 ↵

其中， G 格式表示无断点连续运行程序， GB 格式表示带断点连续运行程序。连续运行过程中，当遇到断点或按下 Ctrl+C 键时，终止程序运行。

I 从I/O端口读入数据并显示

格式： I[I/O 接口地址] ↵

如： >I 0042 显示地址为 0042 接口单元的内容。

M 数据块搬移

格式： M[段址:]源起始地址， [段址:]目标起始地址 ↵

O 数据送存指定I/O接口地址单元

格式： O I/O 接口地址 数据 ↵

如： >O 0600 80 就完成送 80 到地址为 0600 的 I/O 端口去。

R 寄存器或片内RAM区显示与修改

格式： R ↵

R ↵ 操作后，屏幕显示：CS=XXXX DS=XXXX IP=XXXX AX=XXXX F=XXXX

若需要显示并修改特定寄存器内容，则选择 R 寄存器名 ↵ 操作。如 R AX ↵，则显示： AX=XXXX，键入回车键，结束该命令。若输入四位十六进制数并回车，则将该数填入寄存器 AX 中，并结束该命令。

T 单步运行指定的程序

格式： T[=[段址:]偏移量] ↵

每次按照指定的地址或 IP/PC 指示的地址，单步执行一条指令后则显示运行后的 CPU 寄存器情况。

U 反汇编

格式： U[[段址:]起始地址[, 尾地址]] ↵

U ↵

系统提供反汇编程序能力，上面第一格式可实现连续显示从某地址到另一高端地址间的代码反汇编，而后一种格式每次只能显示当前行。

2.2 系统编程信息

2.2.1 地址分配情况

1. 系统内存分配

系统内存分配情况如图 2-27 所示。系统内存分为程序存储器和数据存储器，程序存储器为一片 128KB 的 FLASH ROM，数据存储器为一片 128KB 的 SRAM。（程序存储器可以扩展到 256KB，数据存储器可以扩展到 512KB）

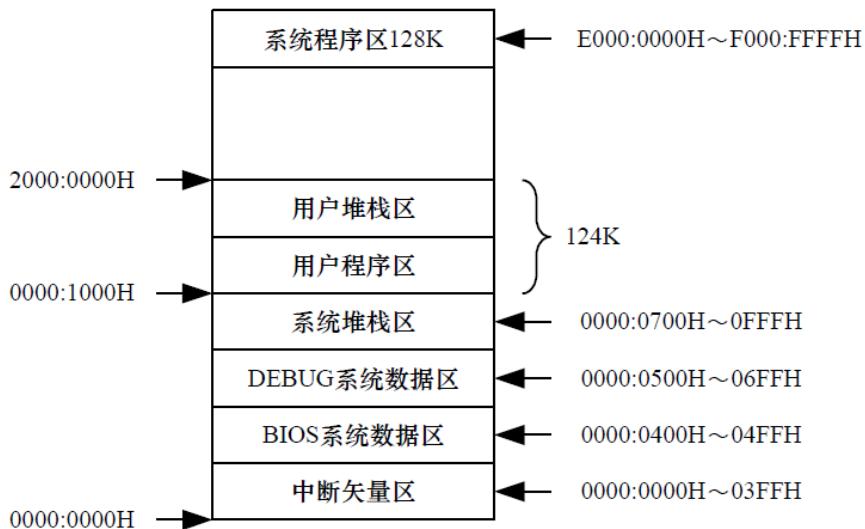


图 2-27 系统内存分配

2. 系统编址

采用内存与 I/O 独立编址形式，内存地址空间和外设地址空间是相对独立的。内存地址是连续的 1M 字节，从 00000H~FFFFFH。外设的地址范围从 0000H~FFFFH，共 64K 字节。

(1) 存储器编址

存储器编址情况如表 2-4 所示。

表 2-4 存储器编址

| | 信号线 | 编址空间 |
|---------|-----|---------------|
| 系统程序存储器 | | E0000H~FFFFFH |
| 系统数据存储器 | | 00000H~1FFFFH |
| 扩展存储器 | MY0 | 80000H~9FFFFH |
| | MY1 | A0000H~BFFFFH |

即 SRAM 空间: 00000H~1FFFFH 共 128K

其中: 00000H~00FFFFH 为 4K 系统区

01000H~1FFFFH 为 124K 用户使用区

FLASH 空间: 0E0000H~0FFFFFFH 共 128K

其中: 0E0000H~0EFFFFH 为 64K 用户使用区

0F0000H~0FFFFFFH 为 64K 系统监控区

(2) 输入/输出接口编址

输入/输出接口编址如表 2-5 所示。

表 2-5 输入/输出接口编址

| | 信号线 | 编址空间 |
|-----------|------|-------------|
| 主片 8259 | | 20H、21H |
| 从片 8259 | | A0H、A1H |
| 扩展 I/O 接口 | IOY0 | 0600H~063FH |
| | IOY1 | 0640H~067FH |
| | IOY2 | 0680H~06BFH |
| | IOY3 | 06C0H~06FFH |

2.2.2 常用 BIOS 及 DOS 功能调用说明

表 2-6 INT 03H 使用说明

| |
|----------|
| 入口: 无 |
| 功能: 程序终止 |

表 2-7 INT 10H 使用说明

| |
|---------------------------------------|
| 入口: AH=01H, AL=数据 |
| 功能: 写 AL 中的数据到屏上 |
| 入口: AH=06H, DS: BX=字串首址, 且字串尾用 00H 填充 |
| 功能: 显示一字串, 直到遇到 00H 为止 |

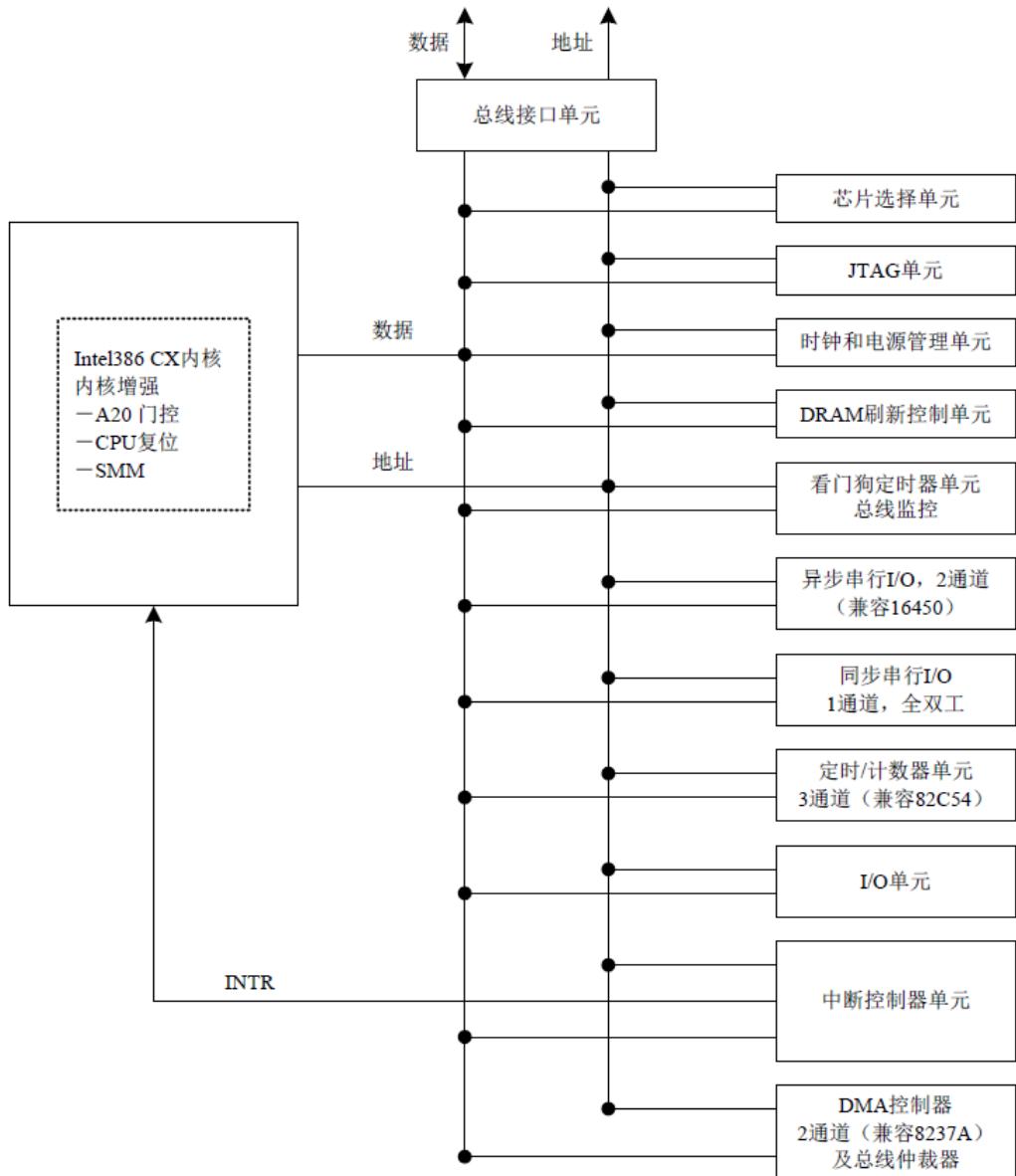
表 2-8 INT 16H 使用说明

| |
|---|
| 入口: AH=00H |
| 功能: 读键盘缓冲到 AL 中, 读指针移动, ZF=1 无键值, ZF=0 有键值 |
| 入口: AH=01H |
| 功能: 检测键盘缓冲, 并送到 AL 中, 读指针不动, ZF=1 无键值, ZF=0 有键值 |

表2-9 INT 21H 使用说明

| |
|---|
| 入口: AH=00H 或 AH=4CH |
| 功能: 程序终止 |
| 入口: AH=01H |
| 功能: 读键盘输入到 AL 中并回显 |
| 入口: AH=02H, DL=数据 |
| 功能: 写 DL 中的数据到显示屏 |
| 入口: AH=08H |
| 功能: 读键盘输入到 AL 中无回显 |
| 入口: AH=09H, DS:DX=字符串首地址, 字符串以 '\$' 结束 |
| 功能: 显示字符串, 直到遇到 '\$' 为止 |
| 入口: AH=0AH, DS:DX=缓冲区首地址, (DS:DX)=缓冲区最大字符数, (DS:DX+1)=实际输入字符数, (DS:DX+2)=输入字符串起始地址 |
| 功能: 读键盘输入的字符串到 DS:DX 指定缓冲区中并以回车结束 |

附录 1 Intel 386TMEX 嵌入式处理器框图

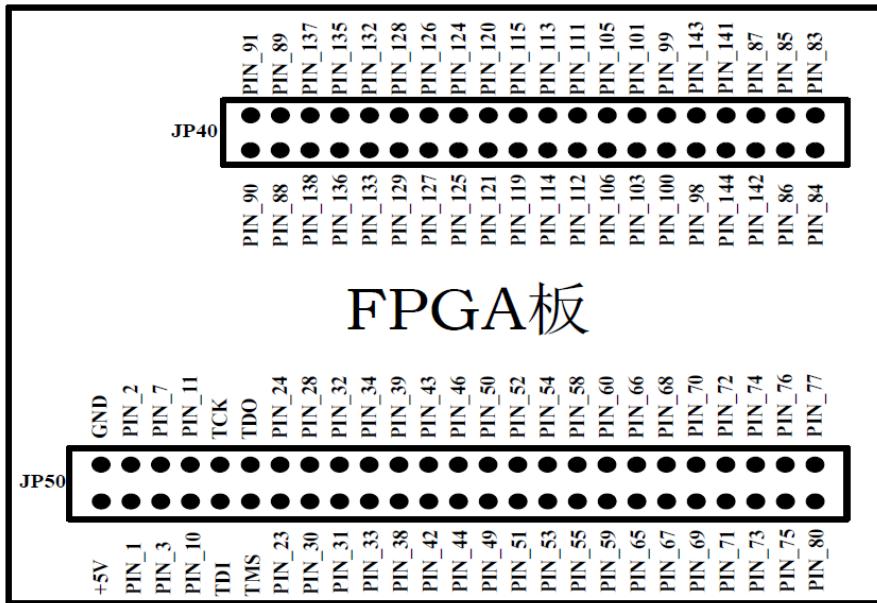


附图 1 Intel 386TMEX 嵌入式处理器框图

附录 2 FPGA 板介绍

FPGA 板有一个 50MHz 有源晶振与芯片的 PN_25 脚(CLK3)连接，可用作全局时钟信号。FPGA 板的“JTAG 下载”口在 8251 单元和 8237 单元中间，可供用户调试下载。

FPGA 板上扩展插座的引脚分配如附图 2 所示。



附图 2 FPGA 接口引脚分配图

出厂时FPGA 板提供的逻辑构成说明

对于微机系统与接口技术实验系统，在出厂时已经为FPGA板配有接口芯片逻辑，可以满足所需的实验要求，它的逻辑电路构成主要有：8259 单元、8255 单元、8254 单元、8251 单元和 8237 单元。各接口芯片和 FPGA 板引脚的对应关系如附表 1 所示。

附表 1 FPGA 板引脚对应关系

| FPGA引脚 | 功能 | 备注 | FPGA引脚 | 功能 | 备注 |
|--------|-----|-------------|--------|-----|--------------|
| 1 | PA7 | 8255 A 口 | 83 | A15 | 各接口芯片 地址线 |
| 3 | PA6 | | 85 | A14 | |
| 10 | PA5 | | 87 | A13 | |
| 31 | PA4 | | 141 | A12 | |
| 33 | PA3 | | 143 | A11 | |
| 38 | PA2 | | 99 | A10 | |
| 42 | PA1 | | 101 | A9 | |
| 44 | PA0 | | 105 | A8 | |
| 49 | PB7 | 8255 B 口 | 111 | A7 | 各接口芯片 数据线 |
| 51 | PB6 | | 113 | A6 | |
| 53 | PB5 | | 115 | A5 | |
| 55 | PB4 | | 120 | A4 | |
| 59 | PB3 | | 124 | A3 | |
| 65 | PB2 | | 126 | A2 | |
| 67 | PB1 | | 128 | A1 | |
| 69 | PB0 | | 132 | A0 | |
| 11 | PC7 | 8255 C 口 | 119 | D7 | 各接口芯片 数据线 |
| 32 | PC6 | | 121 | D6 | |
| 34 | PC5 | | 125 | D5 | |
| 39 | PC4 | | 127 | D4 | |
| 43 | PC3 | | 129 | D3 | |
| 46 | PC2 | | 133 | D2 | |
| 50 | PC1 | | 136 | D1 | |
| 52 | PC0 | | 138 | D0 | |

附表 1 FPGA 板引脚对应关系 (续表)

| FPGA 引脚 | 功能 | 备注 | FPGA 引脚 | 功能 | 备注 |
|---------|------------|------------------|---------|-----------|--------------|
| 23 | 8255_cs | 8254 引脚分配 | 86 | 8237_cs | 8237 引脚分配 |
| 30 | 8254_cs | | 88 | 8237_clk | |
| 54 | 8254_clk0 | | 114 | 8237_hlda | |
| 58 | 8254_out0 | | 112 | 8237_hold | |
| 60 | 8254_gate0 | | 106 | 8237_ack1 | |
| 66 | 8254_clk1 | | 103 | 8237_ack0 | |
| 68 | 8254_out1 | | 100 | 8237_req1 | |
| 70 | 8254_gate1 | | 98 | 8237_req0 | |
| 24 | 8251_cs | 8251 引脚分配 | 144 | 8237_mwr | 8259 引脚分配 |
| 90 | 8251_clk | | 142 | 8237_mrd | |
| 7 | 8251_txd | | 28 | 8259_cs | |
| 2 | 8251_rxd | | 80 | 8259_int | |
| 137 | 8251_intr | | 77 | 8259_inta | |
| 73 | IOW | 各接口芯片 I/O读写信号 | 75 | 8259_ir1 | |
| 71 | IOR | | 76 | 8259_ir0 | |
| 84 | RST# | 复位 | 25 | sys_clk | 50MHz 输入 |

第三章 实验内容

实验一 8255 并行接口实验

1、实验目的

- (1) 学习利用并行接口芯片 8255 构成并行接口电路的基本方法；
- (2) 熟悉掌握并行接口芯片 8255 的基本性能、硬件连接和初始化编程方法。

2、实验内容

编写一个基本输入输出程序，使 8255 的 A 口为输入，B 口为输出，完成拨动开关到数据灯的数据传输。要求：只要开关拨动，数据灯的显示就发生相应改变。

3、实验原理

并行接口是以数据的字节为单位与 I/O 设备或被控制对象之间传递信息。CPU 和接口之间的数据传送总是并行的，即可以同时传递 8 位、16 位或 32 位等。8255 可编程外围接口芯片是 Intel 公司生产的通用并行 I/O 接口芯片，它具有 A、B、C 三个并行接口，用+5V 单电源供电，能在以下三种方式下工作：方式 0—基本输入/输出方式、方式 1—选通输入/输出方式、方式 2—双向选通工作方式。8255 的内部结构及引脚如图 3-1-1 所示。

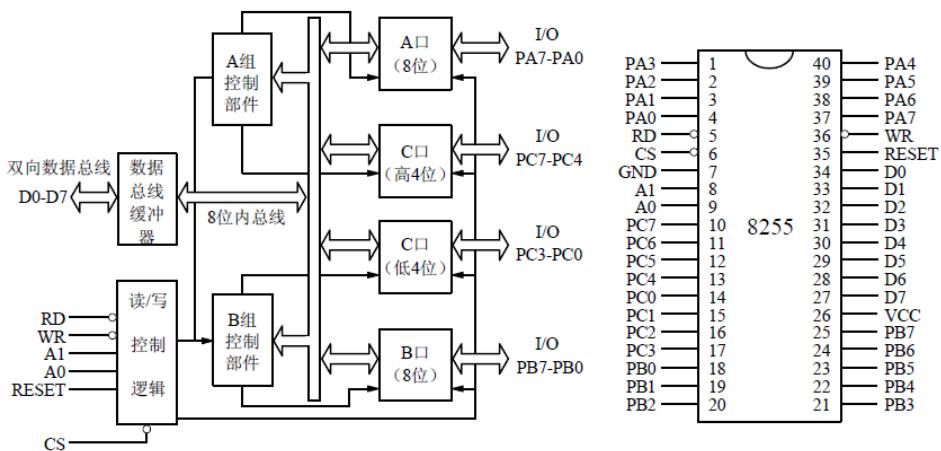


图 3-1-1 8255 内部结构及外部引脚

8255 工作方式控制字和 C 口按位置位/复位控制字格式如图 3-1-2 所示。

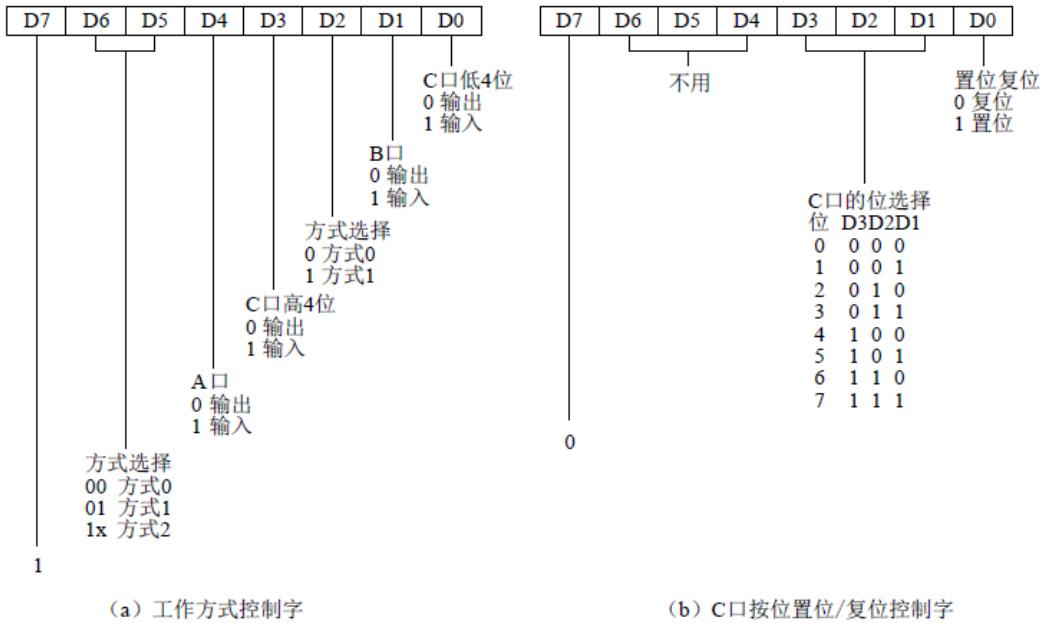


图 3-1-2 8255 控制字格式

8255 实验单元电路如图 3-1-3 所示。

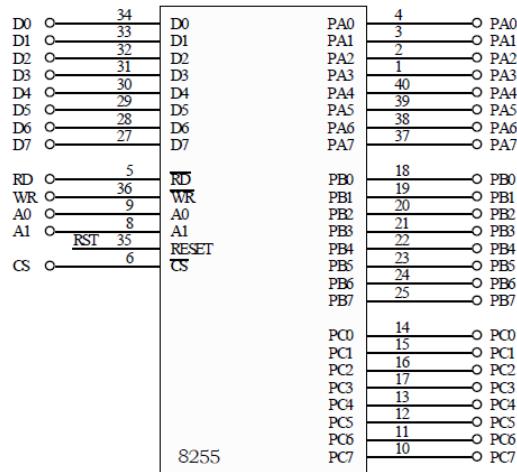


图 3-1-3 8255 实验单元电路图

4、实验步骤

- (1) 关闭实验箱电源。
- (2) 实验接线如图3-1-4 所示，按图连接实验线路。
- (3) 编写实验程序，经编译、链接无误后装入系统。
- (4) 运行程序，拨动开关，同时观察LED显示，验证程序功能。

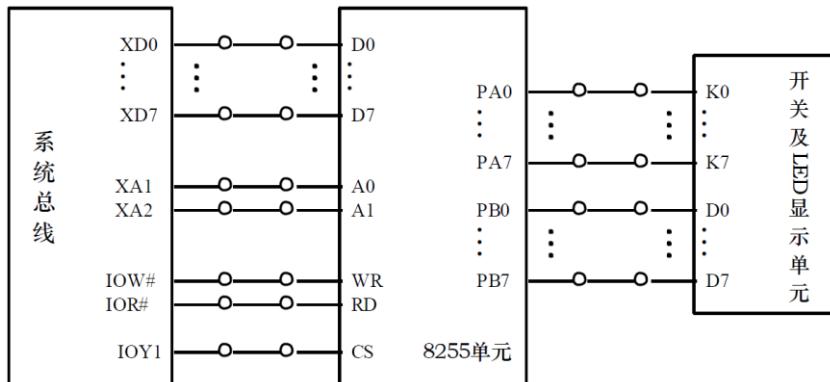


图 3-1-4 实验接线图

5、选做实验

- (1) 在基础上增加如下功能：若开关K7~K0均为高电平，则程序退出。
- (2) 编写程序实现如下功能：当K0为高电平时，数据灯高4位亮；当K0为低电平时，数据灯低4位亮。
- (3) 编写程序实现如下功能：当K1K0=00时，数据灯全部熄灭；当K1K0=01时，数据灯低4位亮；当K1K0=10时，数据灯高4位亮；当K1K0=11时，数据灯全亮。

实验二 存储器扩展实验

1、实验目的

- (1) 了解存储器扩展的方法和存储器的读/写;
- (2) 掌握 CPU 对 16 位存储器的访问方法;
- (3) 通过观测时序图，掌握存储器读写的时序关系。

2、实验内容

按照规则字写存储器。编写实验程序，将 0000H~000FH 共 16 个数写入 SRAM 从 8000:0000H 起始的一段空间中，并通过系统命令查看该存储空间，检查写入数据是否正确。

3、实验原理

存储器是用来存储信息的部件，是计算机的重要组成部分。静态 RAM 是由 MOS 管组成的触发器电路，每个触发器可以存放 1 位信息。只要不掉电，所储存的信息就不会丢失。因此，静态 RAM 工作稳定，不要外加刷新电路，使用方便。但一般 SRAM 的每一个触发器是由 6 个晶体管组成，SRAM 芯片的集成度不会太高，目前较常用的有 6116 (2K×8 位)，6264 (8K×8 位) 和 62256 (32K×8 位)。本实验平台上选用的是 6264，两片组成 8K×16 位的形式，共 16K 字节。6264 的外部引脚图如图 3-2-1 所示。

本系统采用准 32 位 CPU，具有 16 位外部数据总线，即 D0、D1、...、D15，地址总线为 BHE# (#表示该信号低电平有效)、BLE#、A1、A2、...、A20。存储器分为奇体和偶体，分别由字节允许线 BHE# 和 BLE# 选通。

存储器中，**从偶地址开始存放的字称为规则字，从奇地址开始存放的字称为非规则字。**处理器访问规则字只需要一个时钟周期，BHE#和 BLE#同时有效，从而同时选通存储器奇体和偶体。处理器访问非规则字却需要两个时钟周期，第一个时钟周期 BHE #有效，访问奇字节；第二个时钟周期 BLE#有效，访问偶字节。处理器访问字节只需要一个时钟周期，视其存放单元为奇或偶，而 BHE#或 BLE#有效，从而选通奇体或偶体。写规则字和非规则字的简单时序图如图 3-2-2 所示。

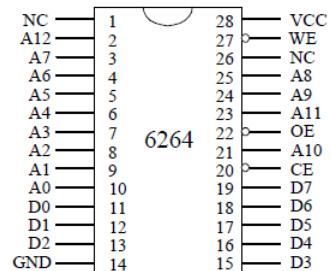


图 3-2-1 6264 引脚图

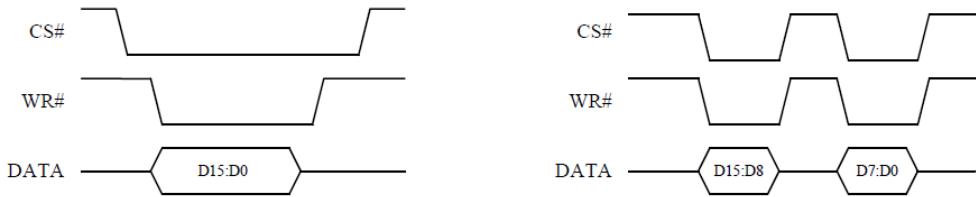


图 3-2-2 写规则字（左）和非规则字（右）简单时序图

实验单元电路如图 3-2-3 所示。

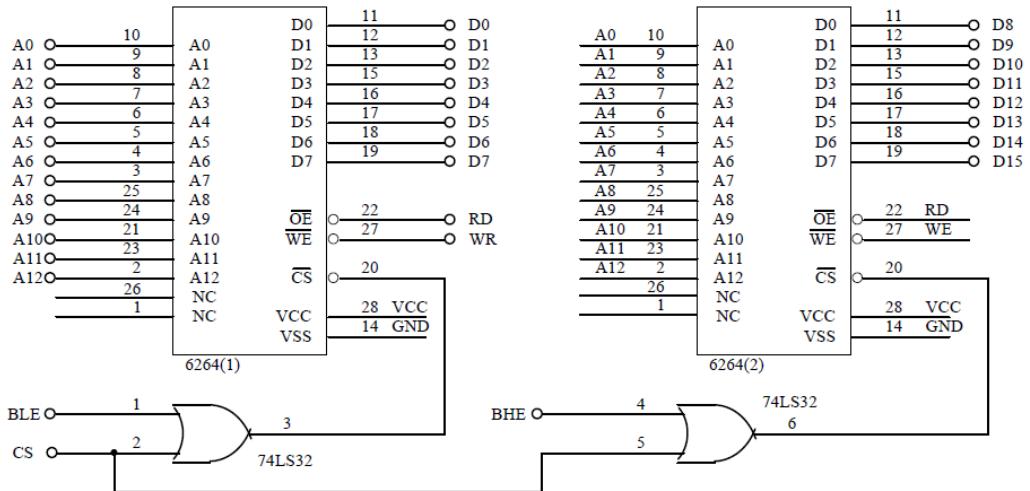


图 3-2-3 实验单元电路图

4、实验步骤

（注：本实验选择 16 位寄存器）

- (1) 关闭实验箱电源。
- (2) 实验接线如图3-2-4所示，按图接线。

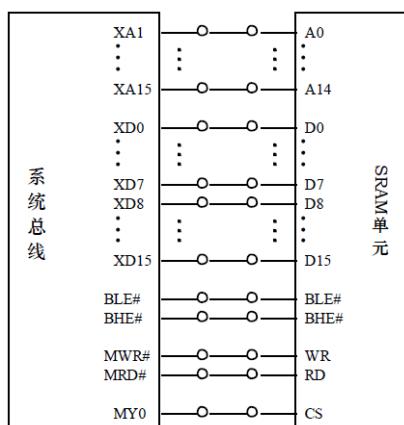


图3-2-4 实验接线图

(3) 编写实验程序，经编译、链接无误后装入系统。

(4) 单击 按钮打开“时序观测窗”，在窗内右键调出“选择观察信号”界面，勾选要观测的信号“MY0、BHE#、BLE#、MWR#、MRD#、XA19...0、XD15...8、XD7...0”，点击确定。

(5) 鼠标单击程序中“MOV [SI], AX”，单击 ，在此处设置断点。

(6) 点击 程序运行至断点处停止，再单击 单步运行。

(7) 点开“时序观测窗”，显示结果如图3-2-5所示。

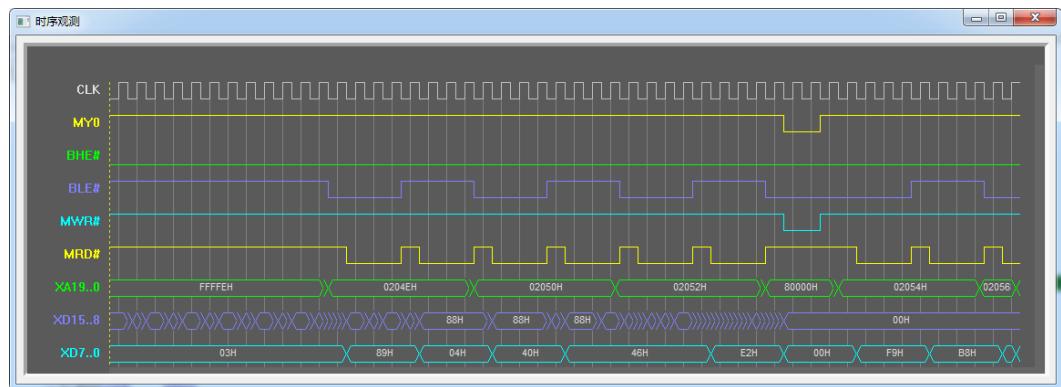


图3-2-5 写规则字时序观测图（1）

(8) 从图中可以看出存储器片选 MY0有效期间，CPU向“8000:0000H”地址进行存储器写操作，写入 00H。

(9) 再点击 程序运行至断点处停止，单击 单步运行。可观察到存储器片选 MY0有效期间，CPU向“8000:0002H”地址进行存储器写操作，写入01H。结果如图3-2-6所示。

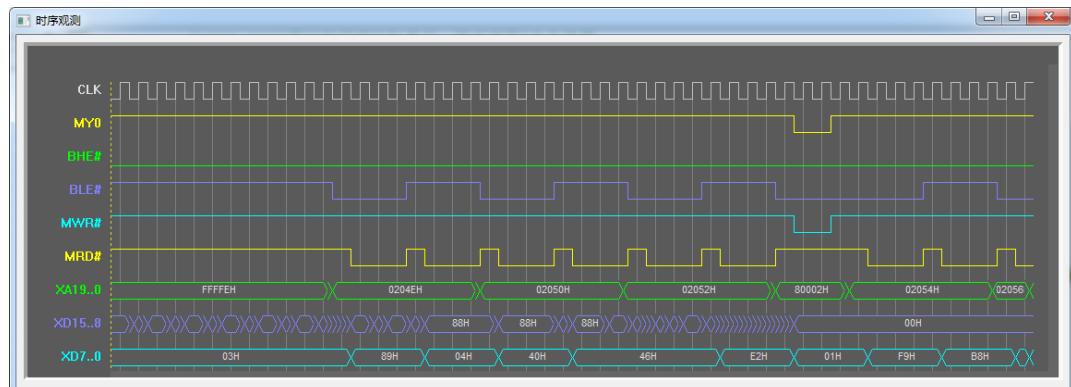


图3-2-6 写规则字时序观测图（2）

(10) 在“Memory”区查询“8000:0000H”地址和“8000:0002H”地址的数据，是否正确。

(11) 单击清除断点，点击运行至程序结束。在“Memory”区查询“8000:0000H”地址至“8000:001FH”地址的数据，是否正确。

6、选做实验

- (1) 改变实验程序，按非规则字写存储器，观察实验结果。
- (2) 改变实验程序，按字节方式写存储器，观察实验结果。

实验三 8259 中断优先级实验

1、实验目的

- (1) 掌握 8259 中断控制器的工作原理;
- (2) 学习 8259 的应用编程方法;
- (3) 掌握 8259 级联方式的使用方法;
- (4) 通过观测时序图，掌握中断请求、中断响应信号及中断矢量的时序关系。

2、实验内容

- (1) 单中断请求实验：单脉冲输出与主片8259的IR7相连，每按动一次单脉冲开关产生一次外部中断，在显示屏幕上输出一个字符“7”。
- (2) 双中断请求实验：主程序通过8255的PB口输出OFFH，使数据灯D0~D7全亮；按单脉冲开关KK1+使绿灯亮（红灯灭），延时一段时间后返回主程序；按单脉冲开关KK2+使红灯亮（绿灯灭），延时一段时间后返回主程序。

3、实验原理

在Intel 386EX芯片中集成有中断控制单元（ICU），**该单元包含有两个级联中断控制器**，一个为主控制器，一个为从控制器。该中断控制单元就功能而言与工业上标准的82C59A是一致的，操作方法也相同。

8259内部带有优先级排队电路，能对8个级中断源实现优先级控制。当一个中断请求IRx被服务时，又可能有后续的中断请求到达，或者其它中断IRy请求到达。发生这种情况时，8259通过优先级控制电路来决定后续中断请求是否可以打断当前的服务。

在TDX-PITE实验系统中，将主控制器的IR6、IR7开放出来供实验使用。主片8259的IR4供系统串口使用。8259的内部连接及外部管脚引出如图3-3-1所示。

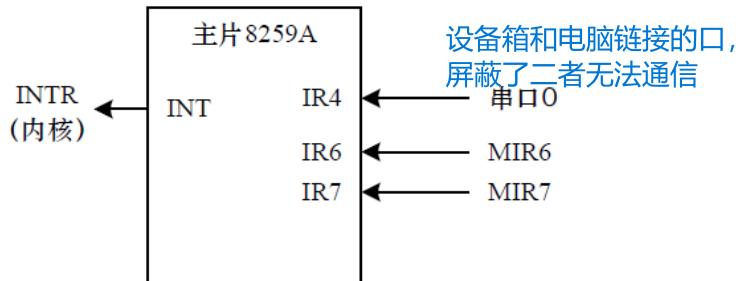


图 3-3-1 8259 内部连接及外部管脚引出图

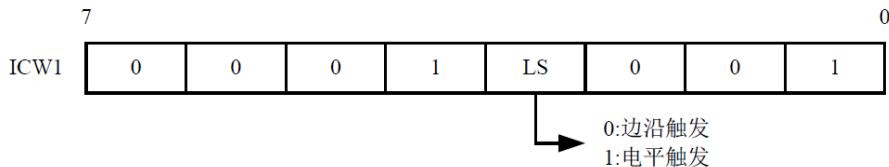
在对8259进行编程时，首先必须进行初始化。一般先使用CLI指令将所有的可屏蔽中断禁止，然后写入初始化命令字。8259有一个状态机控制对寄存器的访问，不正确的初始化顺序会造成异常初始化。**在初始化主片8259时，写入初始化命令字的顺序是：ICW1、ICW2、ICW3、然后是ICW4，初始化从片8259的顺序与初始化主片8259的顺序是相同的。系统启动时，主片8259已被初始化，且4号中断源（IR4）提供给与PC联机的串口通信使用，其它中断源被屏蔽。**

表3-3-1列出了中断控制单元的寄存器相关信息。

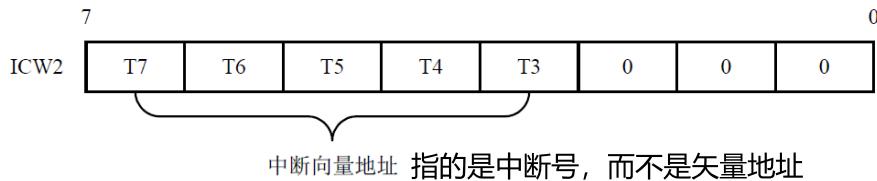
表 3-3-1 ICU 寄存器列表

| 寄存器 | 口地址 | 功能描述 |
|-----------------------------|----------------------------------|--|
| ICW1（主） ICW1（从） （只写） | 0020H 00A0H | 初始化命令字1：决定中断请求信号为电平触发还是边沿触发。 |
| ICW2（主） ICW2（从） （只写） | 0021H 00A1H | 初始化命令字2：包含了8259的基址中断向量号，基址中断向量是IR0的向量号，基址加1就是IR1的向量号，依此类推。 |
| ICW3（主） （只写） | 0021H | 初始化命令字3：用于识别从8259设备连接到主控制器的IR信号，内部的从8259连接到主8259的IR2信号上。 |
| ICW3（从） （只写） | 00A1H | 初始化命令字3：表明内部从控制器级联到主片的IR2信号上。 |
| ICW4（主） ICW4（从） （只写） | 0021H 00A1H | 初始化命令字4：选择特殊全嵌套或全嵌套模式，使能中断自动结束方式。 |
| OCW1（主） OCW1（从） （读/写） | 0021H 00A1H | 操作命令字1：中断屏蔽操作寄存器，可屏蔽相应的中断信号。 |
| OCW2（主） OCW2（从） （只写） | 0020H 00A0H | 操作命令字2：改变中断优先级和发送中断结束命令。 |
| OCW3（主） OCW3（从） （只写） | 0020H 00A0H | 操作命令字3：使能特殊屏蔽方式，设置中断查询方式，允许读出中断请求寄存器和当前中断服务寄存器。 |
| IRR（主） IRR（从） （只读） | 0020H 00A0H | 中断请求：指出挂起的中断请求。 |
| ISR（主） ISR（从） （只读） | 0020H 00A0H | 当前中断服务：指出当前正在被服务的中断请求。 |
| POLL（主） POLL（从） （只读） | 0020H 0021H 00A0H 00A1H | 查询状态字：表明连接到8259上的设备是否需要服务，如果有中断请求，该字表明当前优先级最高的中断请求。 |

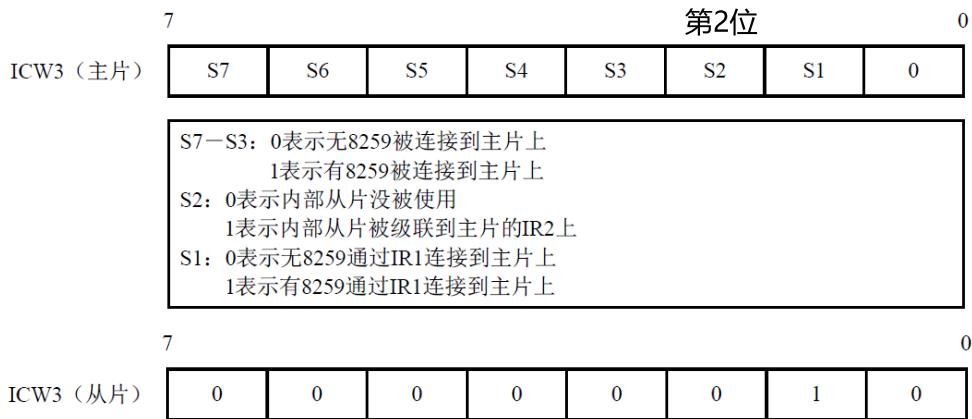
(1) 初始化命令字1 (ICW1) 格式如下:



(2) 初始化命令字2 (ICW2) 格式如下:



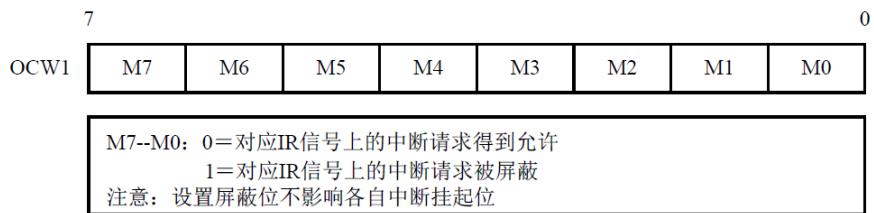
(3) 初始化命令字3 (ICW3) 格式如下:



(4) 初始化命令字4 (ICW4) 格式如下:



(5) 操作命令字1 (OCW1) 格式如下:



(6) 操作命令字2 (OCW2) 格式如下:

| OCW2 | 7 | R | SL | EOI | 0 | 0 | L2 | L1 | L0 | 0 |
|---|---|---|----|-----|---|---|----|----|----|---|
| R SL EOI 命令 | | | | | | | | | | |
| 0 0 0 取消优先级自动循环 * | | | | | | | | | | |
| 0 0 1 一般的中断结束命令 | | | | | | | | | | |
| 0 1 0 无操作 | | | | | | | | | | |
| 0 1 1 特殊的中断结束命令 ** | | | | | | | | | | |
| 1 0 0 中断优先级自动循环 * | | | | | | | | | | |
| 1 0 1 在一般中断结束方式中优先级循环 | | | | | | | | | | |
| 1 1 0 优先级特殊循环方式 ** | | | | | | | | | | |
| 1 1 1 在特殊中断结束方式中优先级循环 ** | | | | | | | | | | |
| * 当8259工作在中断自动结束方式下时, 这些情况可以改变优先级结构。 | | | | | | | | | | |
| ** 在这些情况下优先级由L2:L0指定。 | | | | | | | | | | |
| L2、L1、L0: 在给出特殊的中断结束命令时, L2、L1、L0指出了具体要清除当前中断服务寄存器的哪一位; 当给出特殊的优先级循环方式命令时, L2、L1、L0指出了循环开始时哪个中断的优先级最低。 | | | | | | | | | | |

(7) 操作命令字3 (OCW3) 格式如下:

| OCW3 | 7 | 0 | ESMM | SMM | 0 | 1 | P | RR | RIS | 0 |
|-----------------------|---|---|------|-----|---|---|---|----|-----|---|
| ESMM SMM | | | | | | | | | | |
| 0 0 无影响 | | | | | | | | | | |
| 0 1 无影响 | | | | | | | | | | |
| 1 0 禁止特殊屏蔽模式 | | | | | | | | | | |
| 1 1 使能特殊屏蔽模式 | | | | | | | | | | |
| P: 设置该位使8259工作在中断查询方式 | | | | | | | | | | |
| RR RIS | | | | | | | | | | |
| 0 0 无影响 | | | | | | | | | | |
| 0 1 无影响 | | | | | | | | | | |
| 1 0 读中断请求寄存器IRR | | | | | | | | | | |
| 1 1 读当前中断服务寄存器ISR | | | | | | | | | | |

(8) 查询状态字 (POLL) 格式如下:

| POLL | 7 | INT | - | - | - | - | - | L2 | L1 | L0 | 0 |
|--|---|-----|---|---|---|---|---|----|----|----|---|
| INT: 0=无请求 | | | | | | | | | | | |
| 1=连接在8259上的设备请求服务 | | | | | | | | | | | |
| L2、L1、L0: 当INT为1时, 这些位指出了需要服务的最高优先级的IR; 当INT为0时这些位不确定。 | | | | | | | | | | | |

中断矢量地址与中断号之间的关系如表3-3-2所示。

表 3-3-2 中断矢量地址与中断号之间的关系表

| 主片中断序号 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--------|---------|---------|---------|---------|---------|---------|---------|---------|
| 功能调用 | 08H | 09H | 0AH | 0BH | 0CH | 0DH | 0EH | 0FH |
| 矢量地址 | 20H~23H | 24H~27H | 28H~2BH | 2CH~2FH | 30H~33H | 34H~37H | 38H~3BH | 3CH~3FH |
| 说明 | 未开放 | 未开放 | 未开放 | 未开放 | 串口 | 未开放 | 可用 | 可用 |
| 从片中断序号 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 功能调用 | 30H | 31H | 32H | 33H | 34H | 35H | 36H | 37H |
| 矢量地址 | C0H~C3H | C4H~C7H | C8H~CBH | CCH~CFH | D0H~D3H | D4H~D7H | D8H~DBH | DCH~DFH |
| 说明 | 未开放 | 可用 | 未开放 | 未开放 | 未开放 | 未开放 | 未开放 | 未开放 |

4、实验步骤

- 单中断请求实验

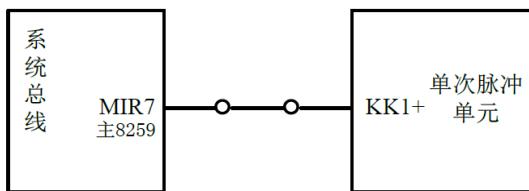


图 3-3-2 8259 单中断实验接线图

- 关闭实验箱电源，按照图3-3-2接线。
- 编写实验程序，经编译、链接无误后装入系统。
- 运行实验程序，重复按单次脉冲开关KK1+，在界面的输出区会显示字符“7”，说明响应了中断。实验现象结果如图3-3-3所示。

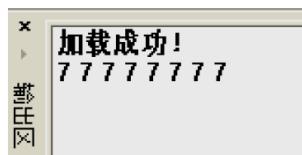


图 3-3-3 8259 单中断实验结果图

中断是有优先级的，0最高，之后递减。

中断嵌套不考。

- 双中断请求实验

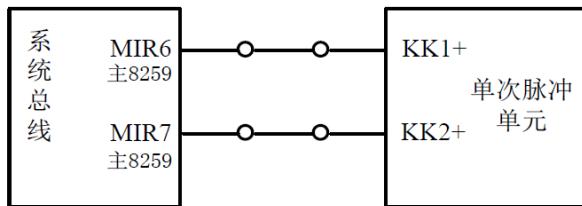


图 3-3-4 8259 双中断实验接线图

- (1) 关闭实验箱电源，按照图3-3-4接线（8255实验单元接线同实验一）。
- (2) 编写实验程序，经编译、链接无误后装入系统。
- (3) 运行实验程序，观察实验现象。

5、选做实验

使用单脉冲开关控制LED灯的移动方向：

- 初始时D7点亮，其余灯熄灭；
- 任意时刻按下KK1，点亮的LED灯立即向右循环移动；
- 任意时刻按下KK2，点亮的LED灯立即向左循环移动。

实验四 A/D 转换实验

1、实验目的

- (1) 学习理解模/数信号转换的基本原理;
- (2) 掌握模/数转换芯片 ADC0809 的使用方法。

2、实验内容

编写实验程序，将 ADC 单元中提供的 0V~5V 信号源作为 ADC0809 的模拟输入量，进行 A/D 转换，转换结果通过变量进行显示。

3、实验原理

ADC0809 包括一个 8 位的逐次逼近型的 ADC 部分，并提供一个 8 通道的模拟多路开关和联合寻址逻辑。用它可直接输入 8 个单端的模拟信号，分时进行 A/D 转换，在多点巡回检测、过程控制等应用领域中使用非常广泛。ADC0809 的主要技术指标为：

- 分辨率：8 位
- 单电源：+5V
- 总的不可调误差：±1LSB
- 转换时间：取决于时钟频率
- 模拟输入范围：单极性 0~5V
- 时钟频率范围：10KHz~1280KHz

ADC0809 的外部管脚如图 3-4-1 所示，地址信号与选中通道的关系如表 3-4-1 所示。模/数转换单元电路图如图 3-4-2 所示。

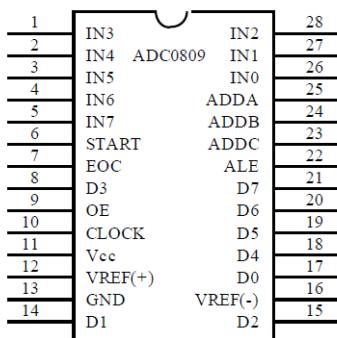


图 3-4-1 ADC 0809 外部引脚

表 3-4-1 地址信号与选中通道的关系

| 地 址 | | | 选中通道 |
|-----|---|---|------|
| C | B | A | |
| 0 | 0 | 0 | IN0 |
| 0 | 0 | 1 | IN1 |
| 0 | 1 | 0 | IN2 |
| 0 | 1 | 1 | IN3 |
| 1 | 0 | 0 | IN4 |
| 1 | 0 | 1 | IN5 |
| 1 | 1 | 0 | IN6 |
| 1 | 1 | 1 | IN7 |

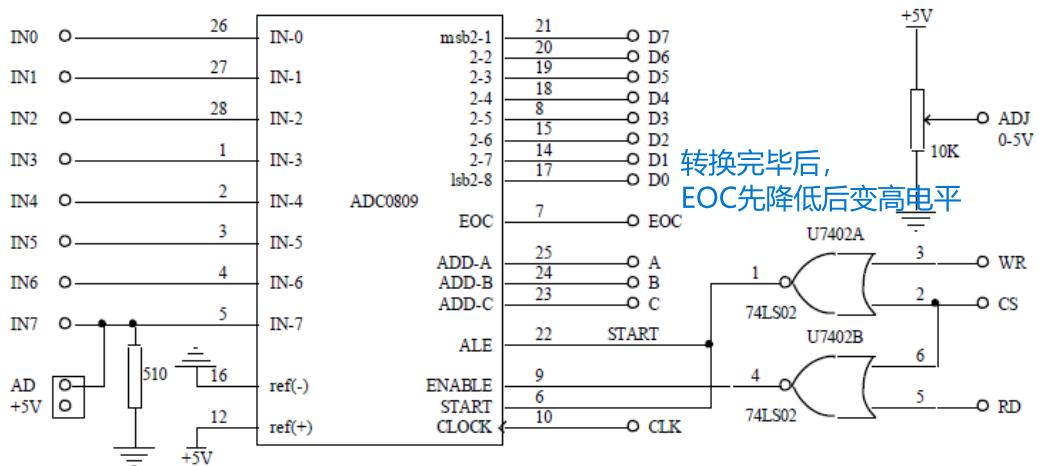


图 3-4-2 模/数转换电路图

4、实验步骤

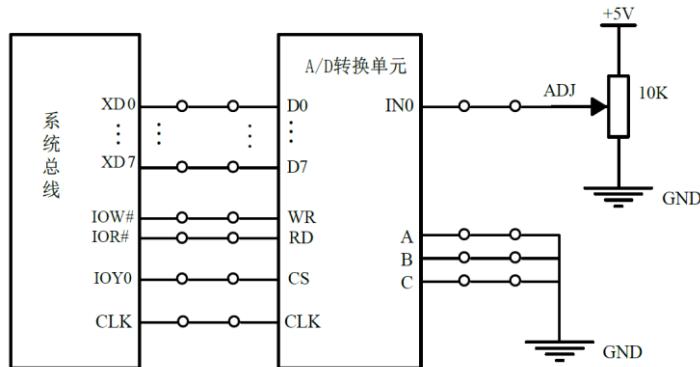


图 3-4-3 A/D 转换实验接线图

- (1) 关闭实验箱电源，按照图 3-4-3 接线。
- (2) 编写实验程序，经编译、链接无误后装入系统。
- (3) 将变量“VALUE”添加到变量监视窗口中。方法如下：打开设置\变量监控，出现如图 3-4-4 的界面，选中要监视的变量“VALUE”，单击“加入监视”后确定，就会在软件左侧栏的“变量区”出现该值。
- (4) 在 JMP START 语句行设置断点，然后运行程序。
- (5) 程序运行到断点处停止运行，查看变量窗口中的 VALUE 值。
- (6) 调节电位器旋钮，改变输入电压，查看 VALUE 值（转换结果）的变化，验证程序功能。



图 3-4-4 变量监控图

5、选做实验

分别使用延时等待、查询和中断三种方式实现本实验的实验内容，并将转换结果通过 8 位 LED 进行显示。

实验五 D/A 转换实验

1、实验目的

- (1) 学习理解数/模转换的基本原理;
- (2) 掌握数/模转换芯片的使用方法;
- (3) 学习软件模拟示波器的使用方法。

2、实验内容

编写程序，产生周期的锯齿波、矩形波、三角波和阶梯波，经 D/A 转换，显示在软件模拟示波器上。要求：每种波形的周期数、阶梯波的阶梯数可通过参数设置。

3、实验原理

D/A 转换器是一种将数字量转换成模拟量的器件，其特点是：接收、保持和转换的数字信息，不存在随温度、时间漂移的问题，其电路抗干扰性较好。大多数的 D/A 转换器接口设计主要围绕 D/A 集成芯片的使用及配置响应的外围电路。TLC7528 是 8 位、并行、两路电压型输出数字-模拟转换器，其主要性能参数如表 3-5-1 所示，引脚如图 3-5-1 所示。D/A 转换单元实验电路图如图 3-5-2 所示。

表 3-5-1 TLC7528 性能参数

| 性能参数 | 参数值 |
|---------------|------------|
| 分辨率 | 8 位 |
| 单电源 | +5V ~ +15V |
| 线性误差 | 1/2LSB |
| 功耗 (VDD=5V) | 20mW |
| 稳定时间 (VDD=5V) | 100ns |
| 数据输入电平 | 与 TTL 电平兼容 |

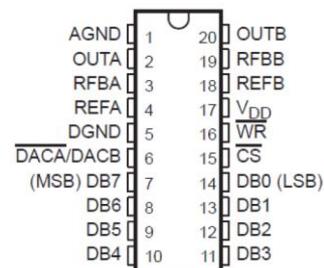


图 3-5-1 TLC7528 外部引脚

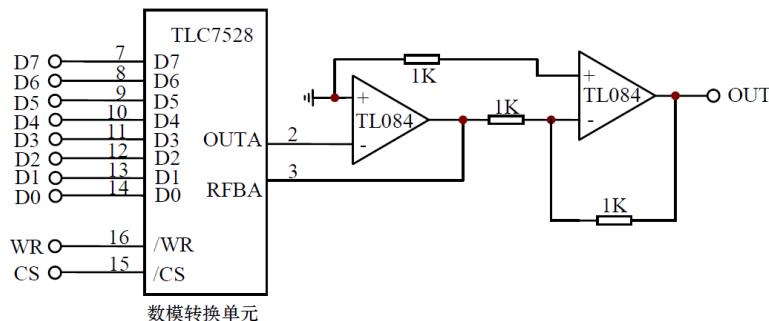


图 3-5-2 D/A 转换实验单元电路图

4、实验步骤

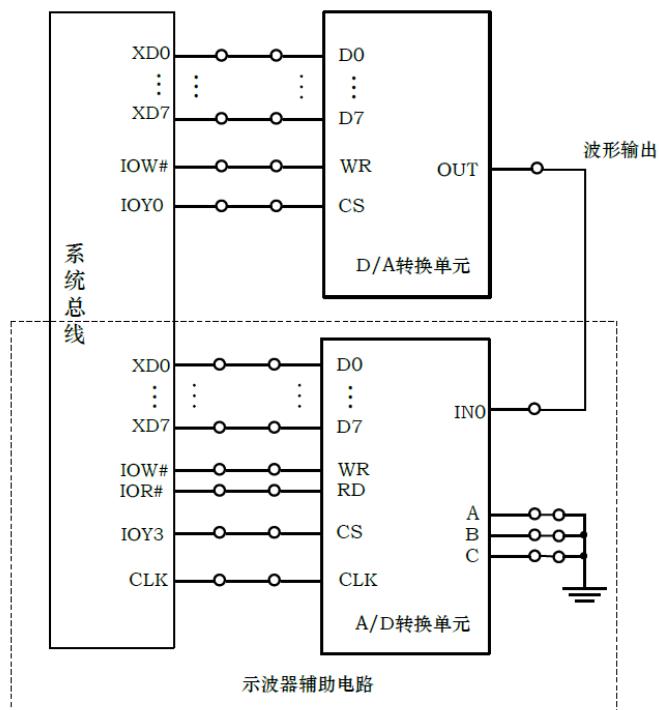


图 3-5-3 D/A 转换实验接线图

- (1) 关闭实验箱电源，按照图 3-5-3 接线。
- (2) 编写实验程序，经编译、链接无误后装入系统。
- (3) 运行程序，用示波器测量 D/A 的输出，观察实验现象。

用示波器观察波形的方法：单击虚拟仪器菜单中的 示波器 按钮或直接单击工具栏的 按钮，在新弹出的示波器界面上单击 按钮运行示波器，观测实验波形。

5、选做实验

在软件模拟示波器上显示锯齿波、矩形波、三角波和阶梯波，并使用单脉冲开关 KK1 在四种波形间进行切换。

实验六 8254 定时/计数器实验

1、实验目的

- (1) 掌握 8254 的工作方式及应用编程;
- (2) 掌握 8254 典型应用电路的接法。

2、实验内容

- (1) 计数应用实验。编写程序，应用 8254 的计数功能，使用单次脉冲模拟计数，使每当按动 ‘KK1+’ 5 次后，产生一次计数中断，并在屏幕上显示一个字符 ‘M’。
- (2) 定时应用实验。编写程序，应用 8254 的定时功能，产生一个 1s 的方波，并使用示波器观察波形。

3、实验原理

8254 是 Intel 公司生产的通用定时/计数器，是 8253 的改进型，比 8253 具有更优良的性能。8254 具有以下基本功能：

- (1) 有 3 个独立的 16 位计数器。
- (2) 每个计数器可按二进制或十进制 (BCD) 计数。
- (3) 每个计数器可编程工作于 6 种不同工作方式。
- (4) 每个计数器允许的最高计数频率为 10MHz。
- (5) 有读回命令，除了可以读出当前计数单元的内容外，还可以读出状态寄存器的内容。
- (6) 计数脉冲可以是有规律的时钟信号，也可以是随机信号。计数初值公式为：

$$n = f_{\text{CLK}_i} / f_{\text{OUT}_i}$$

其中， f_{CLK_i} 是输入时钟脉冲的频率， f_{OUT_i} 是输出波形的频率。

图 3-6-1 是 8254 的内部结构框图和引脚图，它由与 CPU 的接口、内部控制电路和三个计数器组成。8254 的工作方式如下述：

- (1) 方式 0：计数到 0 结束输出正跃变信号方式。
- (2) 方式 1：硬件可重触发单稳方式。
- (3) 方式 2：频率发生器方式。
- (4) 方式 3：方波发生器。
- (5) 方式 4：软件触发选通方式。
- (6) 方式 5：硬件触发选通方式。

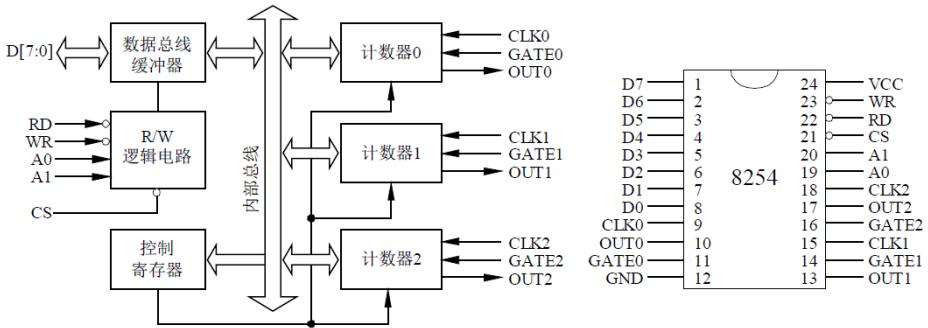


图 3-6-1 8254 内部结构和引脚

8254 的控制字有两个：一个用来设置计数器的工作方式，称为方式控制字；另一个用来设置读回命令，称为读回控制字。这两个控制字共用一个地址，由标识位来区分。8254 控制字和状态字格式如表 3-6-1~表 3-6-3 所示，实验单元电路图如图 3-6-2 所示。

表 3-6-1 8254 方式控制字格式

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----------------|----|---------------------------|----|----------|----|--------|----|
| 计数器选择 | | 读/写格式选择 | | 工作方式选择 | | 计数码制选择 | |
| 00—计数器 0 | | 00—锁存计数值 | | 000—方式 0 | | 0—二进制数 | |
| 01—计数器 1 | | 01—读/写低 8 位 | | 001—方式 1 | | 1—十进制数 | |
| 10—计数器 2 | | 10—读/写高 8 位 | | 010—方式 2 | | | |
| 11—读出控制 字标志 | | 11—先读/写低 8 位 再读/写高 8 位 | | 011—方式 3 | | | |
| | | | | 100—方式 4 | | | |
| | | | | 101—方式 5 | | | |

表 3-6-2 8254 读回控制字格式

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|---------|----------|---------------|----|----|----|
| 1 | 1 | 0—锁存计数值 | 0—锁存状态信息 | 计数器选择（同方式控制字） | 0 | | |

表 3-6-3 8254 状态字格式

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|------------|----------|---------------|----|----|----|----|----|
| OUT 引脚现行状态 | 计数初值是否装入 | 计数器方式（同方式控制字） | | | | | |
| 1—高电平 | 1—无效计数 | | | | | | |
| 0—低电平 | 0—计数有效 | | | | | | |

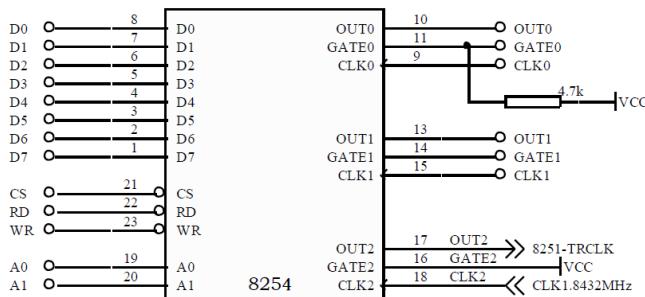


图 3-6-2 8254 实验单元电路图

4、实验步骤

- 计数应用实验

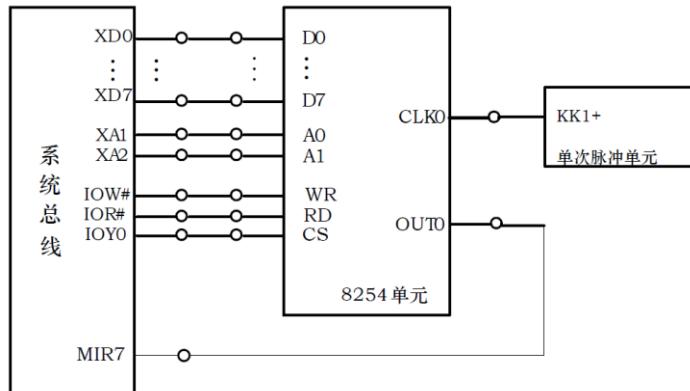


图 3-6-3 8254 计数应用实验接线图

- (1) 关闭实验箱电源，按照图 3-6-3 接线（由于 8254 单元中 GATE0 信号已经上拉 +5V，所以 GATE0 不用接线）。
- (2) 编写实验程序，经编译、链接无误后装入系统。
- (3) 运行实验程序，连续按动 KK1+，观察实验现象。
- (4) 改变计数值，验证 8254 的计数功能。

- 定时应用实验

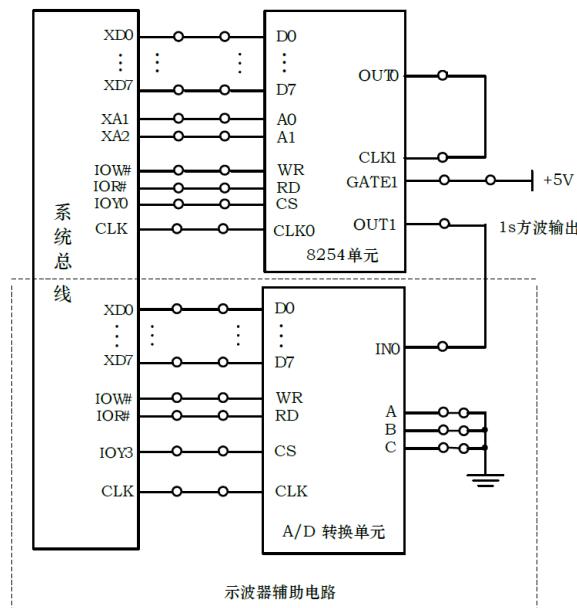


图 3-6-4 8254 定时应用实验接线图

- (1) 关闭实验箱电源，按照图 3-6-4 接线。
- (2) 编写实验程序，经编译、链接无误后装入系统。
- (3) 运行实验程序，8254 的 OUT1 会输出 1s 的方波，用软件模拟示波器观察 OUT1 输出的波形。

5、选做实验

利用 8254（方式 2 或方式 3）精确定时，控制 LED 灯 D7~D0 的点亮：

- 初始时，D7~D0全部熄灭；
- 1秒钟之后点亮D0，再经过1秒D1D0点亮，...，8秒钟之后D7~D0全部点亮。

实验七 8251 串行接口实验

1、实验目的

- (1) 学习串行通信的工作原理和通信协议；
- (2) 掌握串行接口 8251 的硬件连接和编程方法。

2、实验内容

- (1) 数据信号串行传输实验。循环向串口发送一个数，使用示波器测量 TXD 引脚上的波形，以了解串行传输的数据格式。
- (2) 自收自发实验。将 3000H 起始的 10 个单元中的初始数据发送到串口，然后自接收并保存到 4000H 起始的内存单元中。

3、实验原理

- 8251 的基本性能

8251 是可编程的串行通信接口，可以管理信号变化范围很大的串行数据通信。有下列基本性能：

- (1) 通过编程，可以工作在同步方式，也可以工作在异步方式。
- (2) 同步方式下，波特率为 0~64K；异步方式下，波特率为 0~19.2K。
- (3) 在同步方式时，可以用 5~8 位来代表字符，内部或外部同步，可自动插入同步字符。
- (4) 在异步方式时，也使用 5~8 位来代表字符，自动为每个数据增加 1 个启动位，并能够根据编程为每个数据增加 1 个、1.5 个或 2 个停止位。
- (5) 具有奇偶、溢出和帧错误检测能力。
- (6) 全双工，双缓冲器发送和接收器。

注意，8251 尽管通过了 RS-232 规定的基本控制信号，但并没有提供规定的全部信号。

- 8251 的内部结构及外部引脚

8251 的内部结构如图 3-7-1 所示，可以看出，8251 有 7 个主要部分，即数据总线缓冲器、读/写控制逻辑电路、调制/解调控制电路、发送缓冲器、发送控制电路、接收缓冲器和接收控制电路，图中还标识出了每个部分对外的引脚。

8251 的外部引脚如图 3-7-2 所示，共 28 个引脚，每个引脚信号的输入输出方式如图中的箭头方向所示。

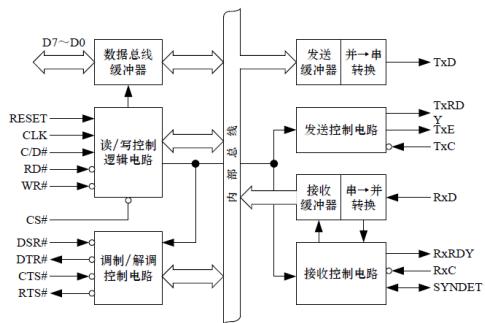


图 3-7-1 8251 内部结构图

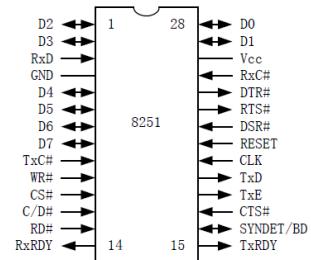


图 3-7-2 8251 引脚图

- 8251 在异步方式下的 TXD 信号上的数据传输格式

图 3-7-3 示意了 8251 工作在异步方式下的 TXD 信号上的数据传输格式。数据位与停止位的位数可以由编程指定。

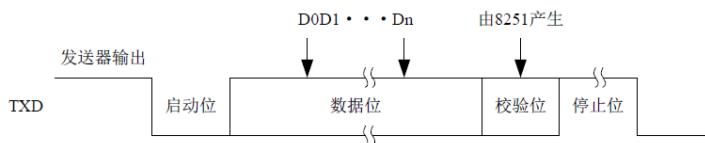


图 3-7-3 8251 工作在异步方式下 TXD 信号的数据传输格式

- 8215 的编程

对 8251 的编程就是对 8251 的寄存器进行操作，下面分别给出 8251 的几个寄存器的格式。

(1) 方式控制字

方式控制字用来指定通信方式及其方式下的数据格式，具体各位定义如下。

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|--|---|----------------------------|----------------------------------|--|--|----|-------|
| SCS/S2 | ESD/S1 | EP | PEN | L2 | L1 | B2 | B1 |
| 同步/停止位 | | 奇偶校验 | | | 字符长度 | | 波特率系数 |
| 同步 (D1D0=00) X0=内同步 X1=外同步 0X=双同步 1X=单同步 | 异步 (D1D0≠0) 00=不用 01=1位 10=1.5位 11=2位 | X0=无校验 01=奇校验 11=偶校验 | 00=5位 01=6位 10=7位 11=8位 | 异步 00=不用 01=01 10=16 11=64 | 同步 00=同步 01=01 10=16 11=64 | | |

(2) 命令控制字

命令控制字用于指定 8251 进行某种操作（如发送、接收、内部复位和检测同步字符串等）或处于某种工作状态，以便接收或发送数据。8251 命令控制字各位定义如下。

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----------------|--------------------------|----------------------|------------------------------|---------------------|----------------------|-------------------------|----------------------|
| EH | IR | RTS | ER | SBRK | RxE | DTR | TxEN |
| 进入搜索 1=允许搜索 | 内部复位 1=使 8251 返回方式控制字 | 请求发送 1=使 RTS 输出 0 | 错误标志复位 使 错误标志 PE、OE、FE 复位 | 发中止字符 1=使 TXD 为低 | 接收允许 1=允许 0=禁止 | 数据终端准备好 1=使 DTR 输出 0 | 发送允许 1=允许 0=禁止 |

(3) 状态字

CPU 通过状态字来了解 8251 当前的工作状态，以决定下一步的操作，8251 的状态字格式如下。

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|--------------------------------|--------|--|---|--------------------------|------|-----------------------|-------------------|
| DSR | SYNDET | FE | OE | PE | TxE | RxDY | TxDY |
| 数据装置就绪： 当 DSR 输入为 0 时，该位为 1 | 同步检测 | 帧错误：该标志仅用于异步方式，当在一字符的结尾没有检测到有效的停止位时，该位置 1。此标志由命令控制字中的位 4 复位。 | 溢出错误：在下一个字符变为可用前，CPU 没有把字符读走，此标志置 1。此错误出现时上一个字符已丢失。 | 奇偶错误： 当检测到奇偶错误时此位置 1。 | 发送器空 | 接收就绪为 1 表明接收缓冲器到一个字符。 | 发送就绪为 1 表明发送缓冲器空。 |

(4) 8251 初始化

8251 的初始化和操作流程如图 3-7-4 所示。

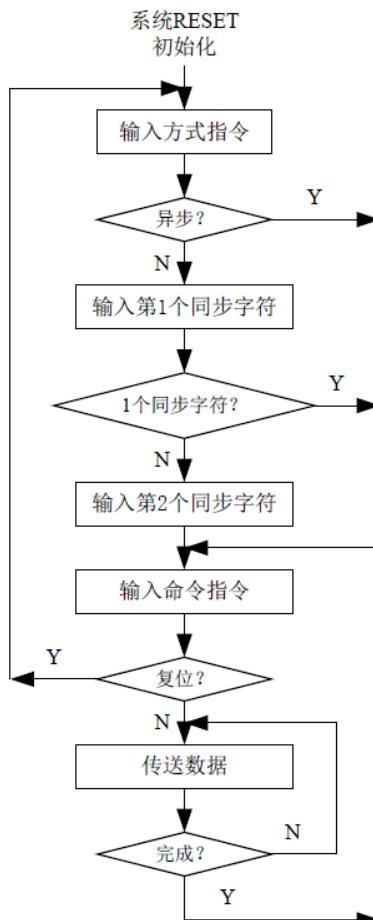


图 3-7-4 8251 初始化流程图

- 8251 实验单元电路

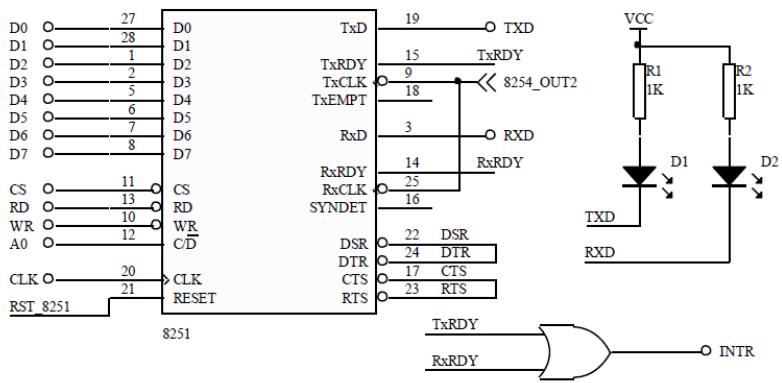


图 3-7-5 8251 实验单元电路图

4、实验步骤

- 数据信号串行传输实验

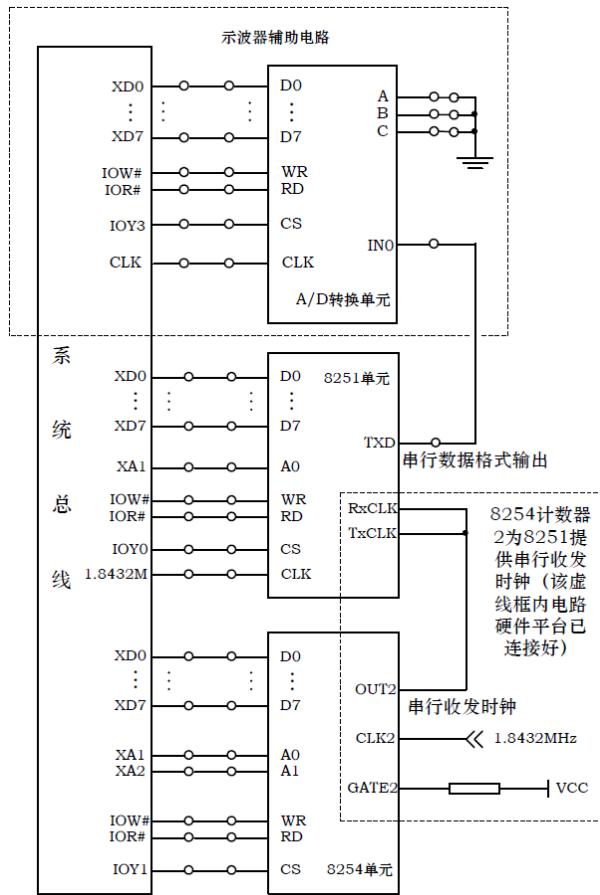


图 3-7-6 数据信号串行传输实验接线图

- (1) 关闭实验箱电源，按照图 3-7-6 接线。
- (2) 编写实验程序，经编译、链接无误后装入系统。
- (3) 运行实验程序，使用软件模拟示波器观察 TXD 引脚上输出的串行格式的数据波形。

• 自收自发实验

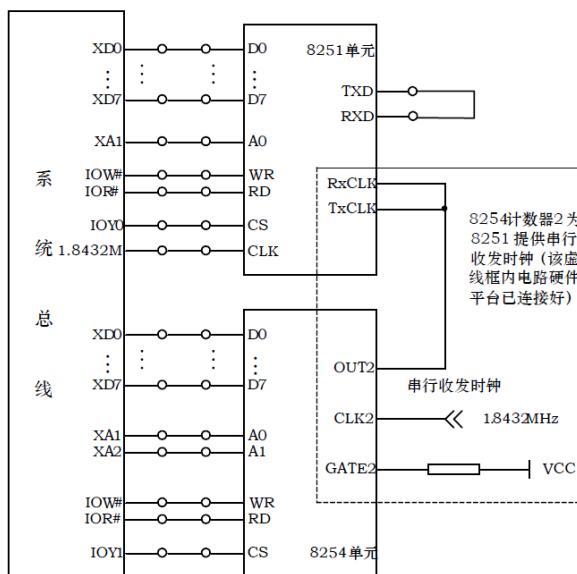


图 3-7-7 自收自发实验接线图

- (1) 关闭实验箱电源，按照图 3-7-7 接线。
- (2) 编写实验程序，经编译、链接无误后装入系统。
- (3) 使用 E 命令更改 3000H 起始的 10 个单元中的数据。
- (4) 运行实验程序，待程序运行停止。
- (5) 查看 4000H 起始的 10 个单元中的数据，与初始化的数据进行比较，验证程序功能。

5、选做实验

编写程序，将字符“a”~“z”的 ASCII 码写入 3000H 开始的存储单元中，然后通过串口以 9600 的波特率传输到 4000H 开始的存储单元中。要求：当出现帧格式错、溢出错和奇偶校验错时，则停止传输，并在屏幕上显示错误码（自行定义）。

实验八 数码管显示实验

1、实验目的

- (1) 学习“七段共阴极数码管”显示的基本原理；
- (2) 掌握通过并行接口芯片控制多个数码管显示的方法。

2、实验内容

用一片 8255 接口芯片的 A 口和 B 口分别连接数码管段码接口（ABCDEFGD_p）和位码接口（X₁~X₆）。编写程序实现以下两种显示方式：

- (1) 使六位数码管从右到左逐位显示移动的数字 0 到 9，即数字 0 从最右端移动到最左端，数字 1 从最右端移动到最左端，……，数字 9 从最右端移动到最左端；循环上述过程。
- (2) 使六位数码管从左到右显示稳定的数字“123456”。

3、实验原理

7 段共阴极数码管的原理图及显示效果如图 3-8-1 所示，其中包含组成字符的 7 个笔画 A~G 以及一个小数点 D_p。7 段共阴极数码管显示 0~9，A~F，其编码依次为：3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H。当显示小数点“.”时在当前编码上置最高位为 1 (80H)。熄灭可用编码 00H，或置位选信号无效。

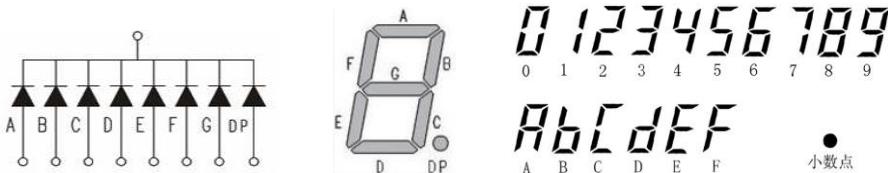


图 3-8-1 七段共阴极数码管原理与效果图

六位数码管和键盘扫描实验单元电路如图 3-8-2 所示。可通过位选信号（X₁~X₆）来选择点亮特定的数码管，并通过输入段码（ABCDEFGD_p）来显示特定数字。对于同时显示多位数字时，可对位码进行扫描。

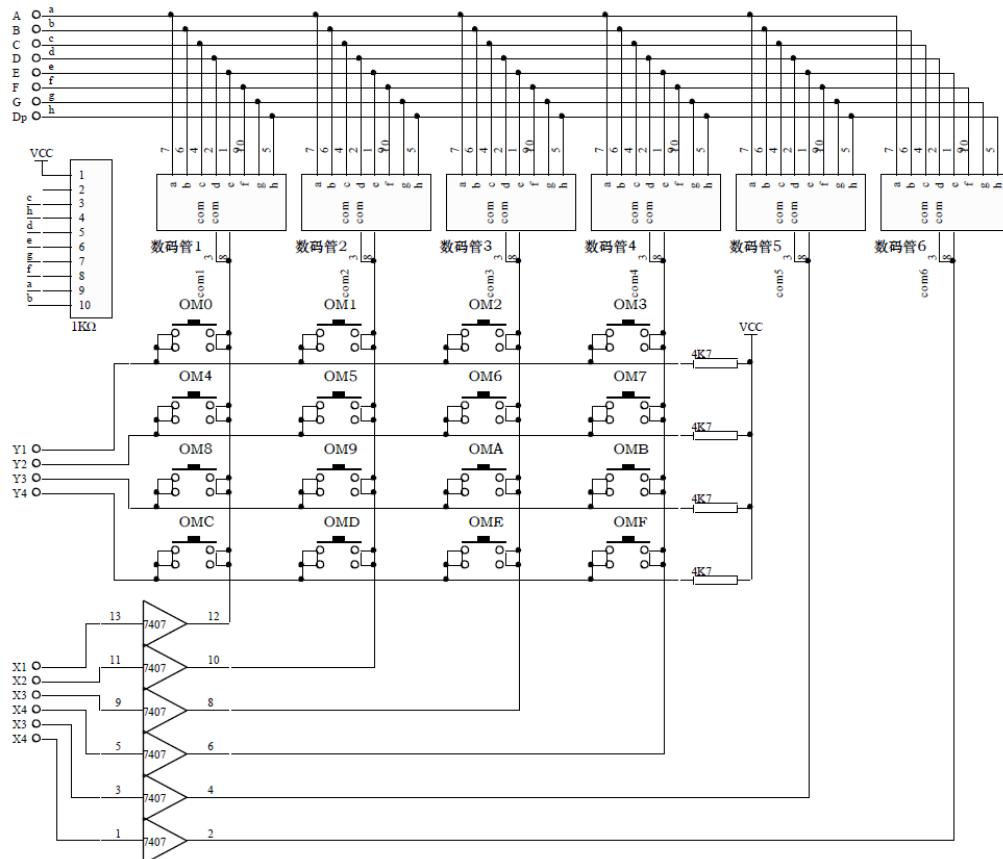


图 3-8-2 键盘扫描及数码管显示实验单元电路

4、实验步骤

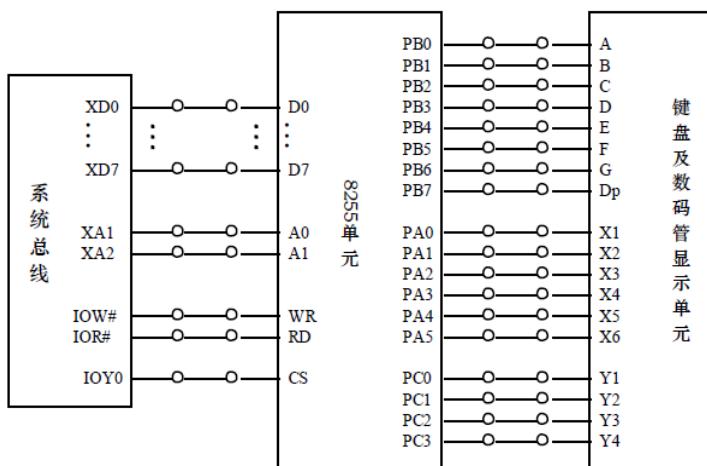


图 3-8-3 数码管显示实验线路图

- (1) 关闭实验箱电源，按照图 3-8-3 接线。
- (2) 编写实验程序，经编译、链接无误后装入系统。
- (3) 运行实验程序，观察实验现象。

5、选做实验

- (1) 在数码管上显示学号后六位，并使用 6 个开关控制对应数码管的亮灭（例如，开关为 1 点亮，为 0 熄灭）。
- (2) 在左侧的三个数码管上显示“123”，每隔 1 秒钟（可用软件延时，也可用 8254 定时）“123”向右移动一位（循环）。

实验九 键盘扫描及数码管显示实验

1、实验目的

- (1) 学习键盘扫描的基本原理。
- (2) 掌握通过并行接口芯片对键盘进行扫描并在多位数码管显示数字的方法。

2、实验内容

编写程序，实现如下功能：初始时数码管无显示；第一次按下键盘时，在最右侧数码管显示对应的十六进制数字；以后每次按下键盘，则将当前显示的数字全部向左移动一位（最左侧的数字移出数码管），并将刚刚键入的数字显示在数码管的最右侧。

3、实验原理

键盘与数码管实验电路请参考实验八中的图 3-8-2。其中键盘对应的数字按照从左到右、从上到下依次为 0~9，A~F。位码接口（X1~X6）除了可以选择点亮特定的数码管以外，还兼 4×4 键盘的列扫描接口。因此实验中需要使用并行接口芯片 8255 兼顾键盘扫描与数码管显示控制。

4、实验步骤

- (1) 关闭实验箱电源，按照实验八中的图 3-8-3 接线。
- (2) 编写实验程序，经编译、链接无误后装入系统。
- (3) 运行实验程序，按下按键，观察数码管的显示，验证程序功能。

5、选做实验

编写程序，将所按键对应的数字（0~E）按要求显示在指定的数码管上：

- 初始时，6 位 LED 数码管上无显示；
- 第一次键入的数字显示在最右端的数码管上；之后每次键入新数字，其显示位置向左移动一位；（只显示最后键入的数字）
- 若当前显示的数字已处于数码管的最左端，此时再键入数字，则将其显示在最左端的数码管上；之后键入数字的显示位置逐次向右移动；
- 如此往复，直至按下数字键 F，程序退出。

实验十 微机接口综合实验

1、实验目的

本实验集成了前面几个实验的内容，是一次综合的复习和提高。

2、实验内容

编写程序实现一个以秒为单位的计时器。

要求：计时时间显示在数码管上；用 8254、8259 和系统时钟产生周期为 1 秒的计时中断；将键盘进行编号，记作 0~F，用来输入计时初值，并控制计时器工作。具体功能如下：

- 初始状态：无显示，此时通过数字键0~9输入计时初值，范围为1~99分钟，显示在数码管1和2上。
- A键（启动/取消）：初值设置完成后按A键，从计时初值开始倒计时，数码管1和2显示分，数码管3和4显示秒；计时过程中按A键，停止倒计时，回到初始状态。
- B键（暂停/继续）：计时过程中按B键，暂停倒计时，显示停止时间；再次按B键，从停止时间继续倒计时。
- 计时结束：倒计时至“0000”时，闪烁三次后回到初始状态。
- C键（退出）：任何时刻按C键，熄灭数码管，程序退出。